Advant® OCS
with Master software

Functional Units Part 7
PIDCONA

User’s Guide

3BSE 002 433R0101

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions.

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant, AdvaCommand, AdvaInform, and AdvaBuild are registered trademarks of ABB Asea Brown Boveri Ltd., Switzerland.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Industrial Systems AB. ABB Industrial Systems AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Industrial Systems AB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Industrial Systems AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Industrial Systems AB's written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

CE MARKING

This product meets the requirements specified in EMC Directive 89/336/EEC and in Low Voltage Directive 72/23/EEC.

Copyright © ABB Industrial Systems AB 1998.

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1 General Information 1-1
1.2 Equipment Requirements 1-1
1.3 Manual Organization 1-1
1.4 Conventions 1-2
1.5 Related Documentation 1-3
1.6 Release History 1-3
1.7 Terminology 1-3
1.8 Product Overview 1-5
1.9 User Interface 1-9

Chapter 2 - Installation

2.1 General 2-1

Chapter 3 - Configuration/Application Building

3.1 Design Considerations 3-1
  3.1.1 Cycle Times 3-1
  3.1.2 Use of Alternative Non-Automatic Modes 3-2
  3.1.3 Use of Alternative Automatic Modes 3-4
  3.1.4 Reverse Action or Direct Action 3-6
  3.1.5 Output Rate Limit and Hard Limits 3-6
  3.1.6 Actuator Position and Direction 3-7
  3.1.7 Scaling and Units 3-7
  3.1.8 Gain Scheduling versus Adaptive Control 3-7
  3.1.9 Precautions in Adaptive Control 3-7
  3.1.10 Feedforward 3-7
  3.1.11 Output Reference Level in Proportional Control 3-8
  3.1.12 Derivative Part Choice 3-8
  3.1.13 Programmed Blocking of Limit Alarms 3-8
  3.1.14 Alternative Event Treatment 3-8
  3.1.15 Deviation in Non-Automatic Modes 3-9
3.2 Capacity and Performance 3-9
3.3 Application Start-up 3-10
3.4 Tutorial 3-10
  3.4.1 Basic Configuration Example 3-10
  3.4.2 Configuration example with Gain Scheduling 3-14
3.5 Application Procedures 3-17
3.6 Configuration/Application Building Menus 3-18

CONTENTS (continued)

Chapter 4 - Runtime Operation

4.1 Product Operation 4-1
    4.1.1 Control Modes 4-1
    4.1.2 Control Algorithm 4-4
    4.1.3 Scaling and Units 4-6
    4.1.4 Setpoint Generation, Ramping, and Tracking 4-7
    4.1.5 Output Generation and Ramping 4-10
    4.1.6 Mode transitions 4-11
    4.1.7 Autotuning 4-13
    4.1.8 Gain Scheduling 4-16
    4.1.9 Adaptive Control 4-18
    4.1.10 Present and Saved Controller Parameters 4-18
    4.1.11 Alarms, Limitations and Events 4-20
4.2 Operating Overview 4-21
4.3 Runtime Tutorial 4-22
4.4 Operating Instructions 4-26
    4.4.1 Change of Control Mode 4-26
    4.4.2 Change of Setpoint or Output 4-27
    4.4.3 Limits 4-28
    4.4.4 Alarms 4-29
    4.4.5 Alarm Blocking 4-30
    4.4.6 Automatic Tuning 4-31
    4.4.7 Manual Tuning 4-37
    4.4.8 Gain Scheduling 4-38
    4.4.9 Adaptive Control 4-40
    4.4.10 Save or Restore Tuned Parameters 4-41
4.5 Runtime Operation Menus 4-41

Chapter 5 - Maintenance

5.1 Preventive Maintenance 5-1
5.2 Hardware Indicators 5-1
5.3 Error Messages 5-1
5.4 Fault Finding and User Repair 5-1
    5.4.1 Invalid Sampling Rate 5-2
    5.4.2 Fault Finding Regarding Autotuning 5-2
    5.4.3 Fault Finding regarding Adaptive Control 5-4

CONTENTS (continued)

Appendix A - Display Elements

A.1 General A-1
  A.1.1 Display Element for Group display: PIDAGR01 .... A-1
  A.1.2 Object Display: PIDAOB01 .... A-3
  A.1.3 Display Element for Process display: PIDAPD01 .... A-7
  A.1.4 Display Element for Process display: PIDAPD02 .... A-7
  A.1.5 Display Element for Process display: PIDAPD03 .... A-8
  A.1.6 Display Element for Process display: PIDAPD51 .... A-9
  A.1.7 Display Element for Process display: PIDAPD52 .... A-10
  A.1.8 Display Element for Process display: PIDAPD53 .... A-11
  A.1.9 Display Element for Object close-up display: PIDAWI01 .... A-12

Appendix B - Process Dialog

B.1 General B-1
B.2 Process Dialog for Adaptive Process Controller PIDCONA B-1
  B.2.1 MIN dialog (No authority key) B-2
  B.2.2 MAX dialog (Authority key) B-4

Appendix C - PC Element

C.1 General C-1

Appendix D - Data Base Elements

D.1 General D-1

Appendix E - Standard Event Texts

E.1 General E-1
E.2 Standard Event Text for Process Controller PIDCONA E-1

Appendix F - Dominant Pole Placement, the PID Tuning Method

F.1 Introduction F-1
F.2 The Basic Idea of Dominant Pole Placement F-1
F.3 Dominant Pole Placement in PIDCONA F-3
F.4 Examples F-4
F.5 Model of the Closed Loop F-7

INDEX

Chapter 1 Introduction

1.1 General Information

The functional unit PIDCONA is one of several functional units available in the Advant OCS process control systems. It is a complete PID control function with autotuning and adaptive control, including functions for operator interface and event handling. This manual describes the functional unit PIDCONA in detail, its functions, how to use it in applications, how to do the configuration, and how to operate it. The purpose is to provide all the information needed for the people who are expected to handle it (control designers, commissioning engineers, operators, service engineers etc.).

1.2 Equipment Requirements

The functional unit PIDCONA requires an Advant Station 500 Series Operator Station, and an Advant Controller 450 with the software library modules QC02-OPF21 and QC02-LIB22 installed. The configuration is done with an Engineering Station in the Advant OCS Series.

1.3 Manual Organization

The *Functional units* is a manual, divided into seven physical parts. Part 1 describes the functional units in general and their common properties. It also introduces the range of functional units available. The other parts describe the individual functional units. This physical item is part 7, describing the PIDCONA Functional Unit. It adheres to the structure, Figure 1-1, generally used for User’s Guides in the Advant OCS systems.

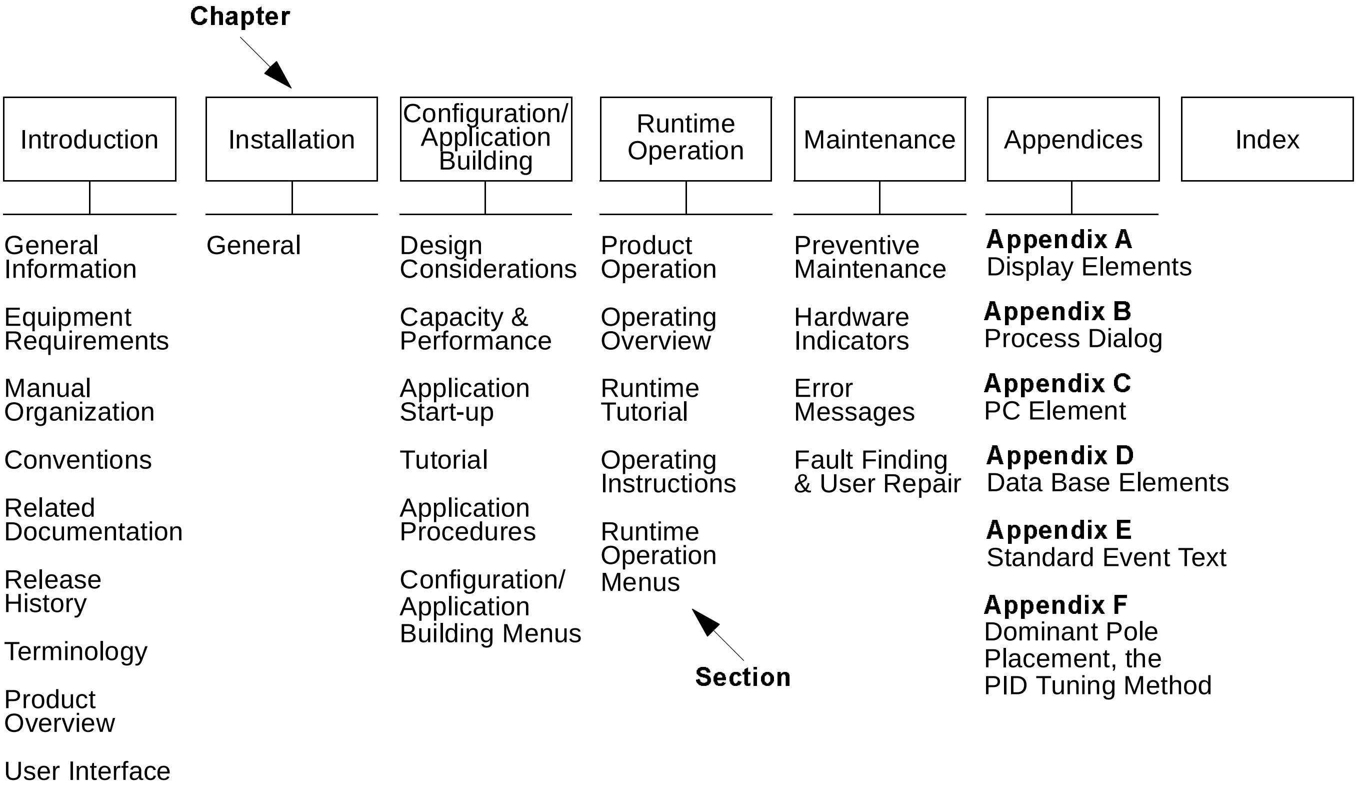


Figure 1-1. The Structure of this Document

The major parts of this manual are Chapter 3, Configuration/Application Building and Chapter 4, Runtime Operation. In Section 4.1, Product Operation, the function of PIDCONA is described, and the rest of Chapter 4, Runtime Operation describes how to operate it. Chapter 3, Configuration/Application Building describes how to apply PIDCONA and how to configure it for the application. Chapter 2, Installation is not applicable.

1.4 Conventions

The word "controller" can mean different things, in this context either the whole Advant Controller 450 or merely the function performing the feedback control in a control loop. The following convention is used, in order to avoid confusion: When 'Controller' is written with initial capital, it refers to the Advant Controller 450, and "controller" written with all lowercase letters refers to a feedback controller, like the PID controller in the functional unit PIDCONA.

1.5 Related Documentation

The common properties of the functional units are described in

- *Functional Units Part 1 Summary and Common Properties*

Application examples for PIDCON (easily adaptable for PIDCONA) appear in

- *ABB MasterPiece 280 Application notes*

The reference information in Appendix A, Display Elements through Appendix E, Standard Event Texts is reproduced from

- *AdvaCommand User Interface Reference Manual*
- *PC elements Advant Controller 400 Series*
- *Data Base elements Advant Controller 400 Series*
- *AdvaCommand Basic Functions User's Guide*

General information regarding operation and configuration is found in

- *AdvaCommand Basic Functions User's Guide*
- *AMPL Configuration Advant Controller 400 Series*
- *AMPL Application Building*

Some manuals, being referred regarding the Operator Station, are

- *AdvaBuild Display Builder User's Guide*
- *AdvaCommand Process Sectioning*
- *AdvaCommand Status List User's Guide*

1.6 Release History

Changes in how the setpoint is displayed in the display elements. The information regarding tuning, tracking, bumpless transfer/mode change, start up and output rate limit has been improved.

1.7 Terminology

Autotuning

Autotuning means a one-shot automatic tuning session started by the user. It leaves the controller tuned for the actual loop in the process, according to the circumstances valid during the tuning. It can be repeated, but always on user demand.

Adaptive Control

In **adaptive control**, while the controller is performing its feedback control task, it also adapts its parameter setting to unforeseen changes of the process dynamics. The difference from autotuning is that the adaptation takes place in continuous operation, without user interaction.

Gain Scheduling

We commonly use the term **gain scheduling** although “parameter scheduling” would in fact be more appropriate. What it means is that different parameter values are used by the controller in different circumstances according to some scheme. What values to use in each situation is controlled by a scheduling variable. In contrast to the case of adaptive control, the setting used in a certain situation is completely predictable from the scheme and its scheduling variable.

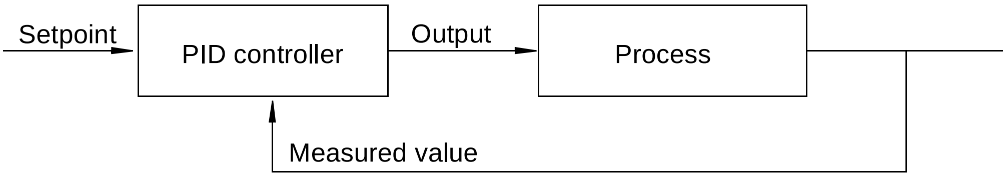


Figure 1-2. Terminology used for basic signals in the controller

Measured Value

The term **measured value** is used for the process output, also called the process variable, the feedback variable, or the controlled variable.

Output

The **output** refers, if nothing else is stated, to the manipulated variable delivered as output from PIDCONA, the controller output, which is also the input to the process.

Setpoint

The goal of the control is to keep the measured value close to the **setpoint** (the desired value, the reference value), in spite of disturbances and varying setpoint values. The “working setpoint” is the present value actually in use.

1.8 Product Overview

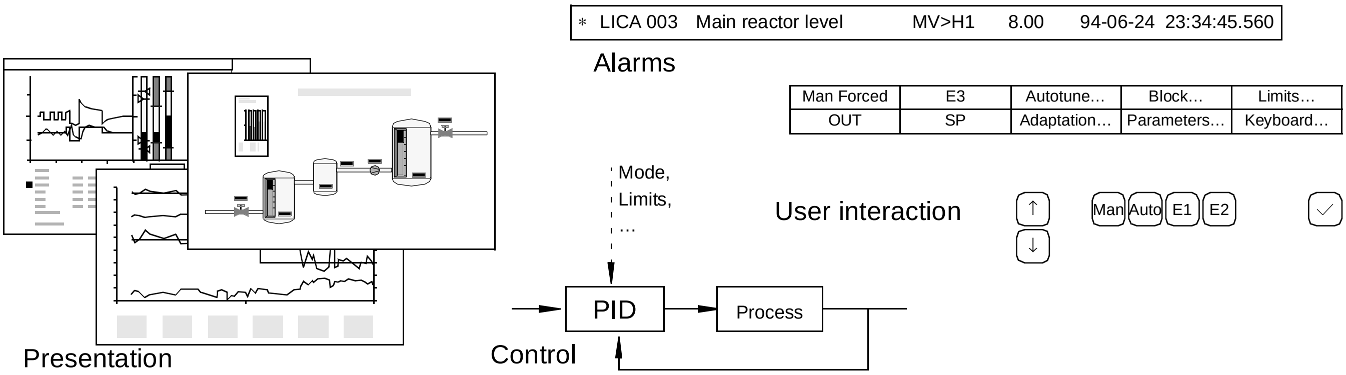
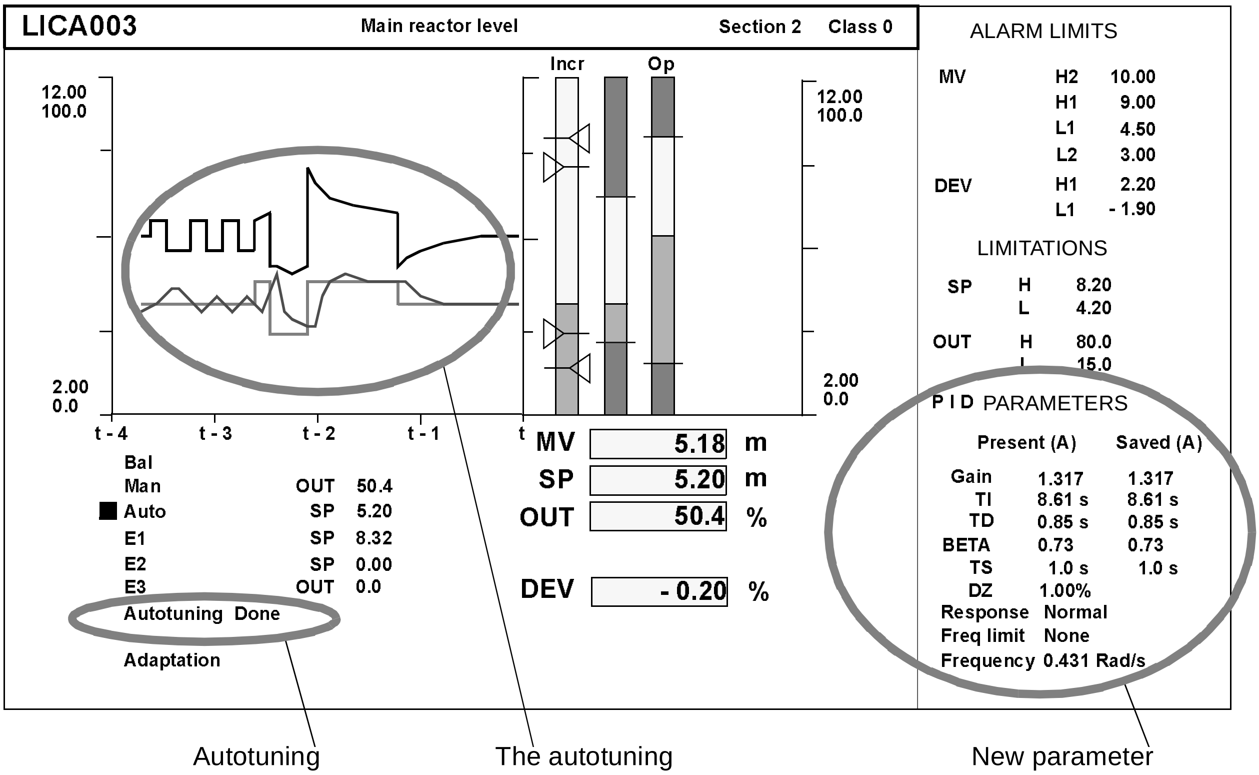


Figure 1-3. Basic functions provided with the functional unit

The functional unit PIDCONA is a standard process controller of the PID class, with a complete operator interface and event handling. In addition, it has a built-in facility to tune itself automatically (autotuning). This autotuning is performed on operator request. It takes the control loop through a sequence of different excitation phases, to gather knowledge about the process and to find and confirm a suitable setting of the controller. The object display shows what you achieved by the autotuning. An example is given in Figure 1-4.



LICA003 Main reactor level Section 2 Class 0

Bal
Man OUT 50.4
■ Auto SP 5.20
E1 SP 8.32
E2 SP 0.00
E3 OUT 0.0
Autotuning Done
Adaptation

MV 5.18 m
SP 5.20 m
OUT 50.4 %
DEV - 0.20 %

ALARM LIMITS

MV	H2	10.00
	H1	9.00
	L1	4.50
	L2	3.00
DEV	H1	2.20
	L1	- 1.90

LIMITATIONS

SP	H	8.20
	L	4.20
OUT	H	80.0
	L	15.0

P I D PARAMETERS

	Present (A)	Saved (A)
Gain	1.317	1.317
TI	8.61 s	8.61 s
TD	0.85 s	0.85 s
BETA	0.73	0.73
TS	1.0 s	1.0 s
DZ	1.00%	
Response	Normal	
Freq limit	None	
Frequency	0.431 Rad/s	

Autotuning is finished

The autotuning sequence performed

New parameter values obtained

Figure 1-4. An object display after completed autotuning

After tuning, the loop is usually controlled without further changes of the obtained setting. In addition to this, PIDCONA can perform gain scheduling and adaptive control.

Gain scheduling is used when the behavior of the process depends on different detectable operating conditions. A set is built by saving the tuned settings for up to five different cases, related to zones of a scheduling variable. As the scheduling variable varies, the setting tuned for its current zone will be used.

Adaptive control means that the controller adapts its setting to unpredicted changes of the behavior of the process. In contrast to autotuning, it is a tuning that goes on continuously without extra excitation. Adaptive control shall be used only when there are unforeseen variations of the process dynamics, large enough to require changes of the setting. It can be combined with gain scheduling. This allows partly predictable larger changes of the process dynamics to be taken care of coarsely by the gain scheduling, leaving the continued fine-tuning to the adaptive control. This is illustrated in Figure 1-5.

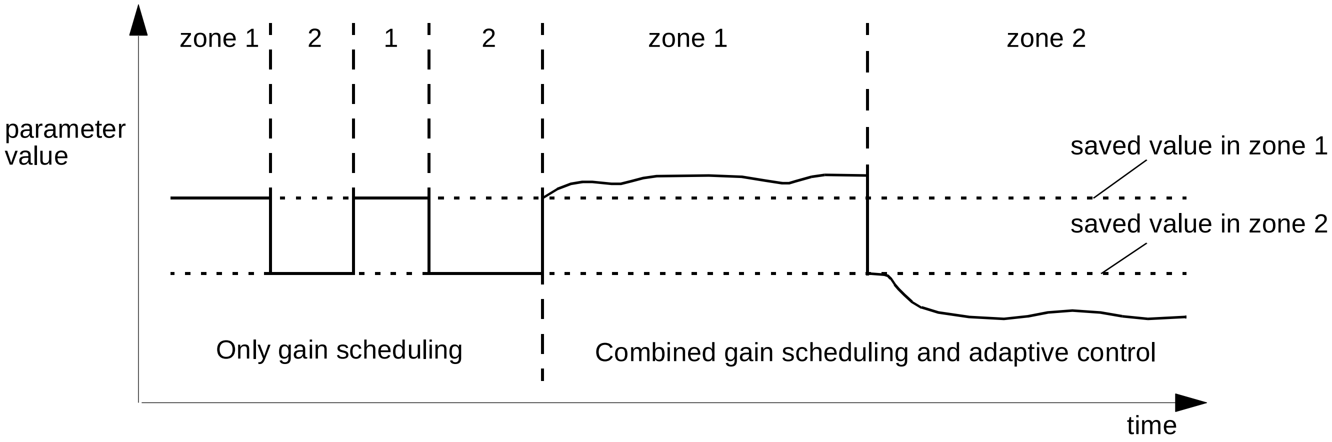


Figure 1-5. Time variation of one of the controller parameters using gain scheduling. The scheduling input alternates between two zones, first without adaptation and then with adaptation.

Both the autotuning and the adaptive control automatically select between PI and PID control. If tuned manually, also pure P or PD control is available.

Control Function

The control modes available for PIDCONA allow different sources for the setpoint during automatic operation, and different sources for the controller output in operation without feedback. The means of selecting the mode, by operator choice or by an input to the PC element or by other events, are systematic with a predefined priority order. This allows schemes like start or stop sequences and interlocking of modes to be implemented in PC programs. The switching between the modes is bumpless, but with an optional possibility to allow “P steps” caused by non-zero control deviation at the moment of starting automatic control. There are also different options for the tracking applied to the operator supplied setpoint, when another setpoint source is used. Rate limits for ramping of setpoint and output changes are built in, with values for different modes obtained from PC element inputs.

The range of modes available makes it simple to configure PIDCONA for easy operation in situations where some different strategies are used interchangeably. Examples of such strategies include ordinary single loop control, cascade control, setpoint optimization, recipe-supplied setpoints, externally supplied control, manual control, and forced or clamped control.

A block scheme illustration of the controller functions is given in Figure 1-6. The setpoint and output generation blocks select their sources according to the present mode. The signal processing block provides filtered values to use for feedback control, back calculations, and adaptation, as required. The relay function substitutes the control algorithm during one phase of autotuning.

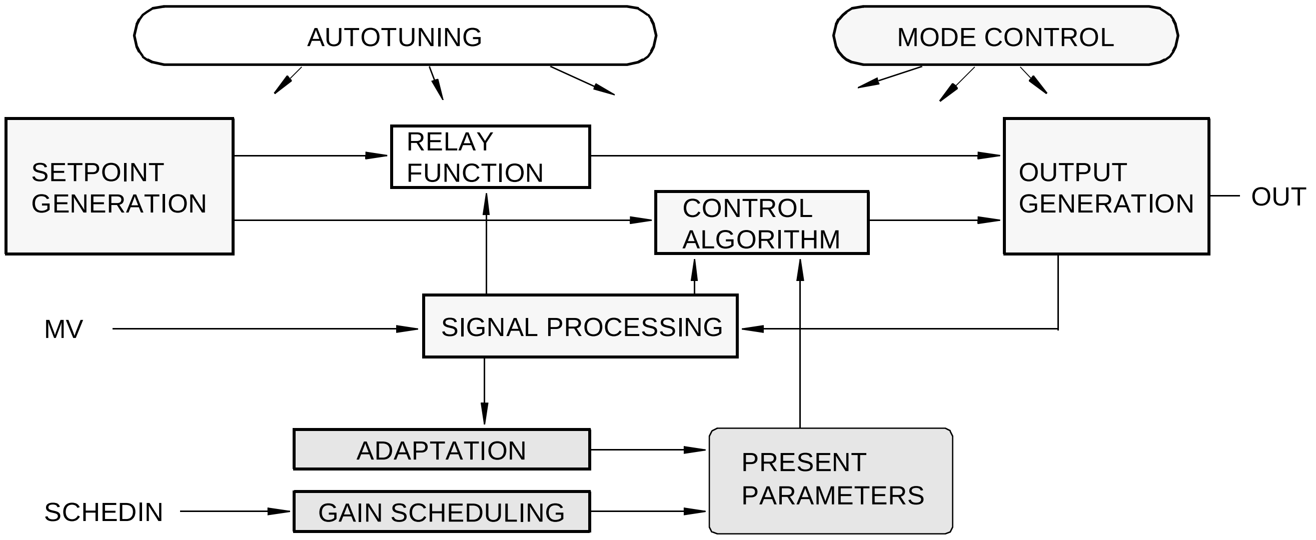


Figure 1-6. Functions involved in the control

Event Handling

Alarm limits, and setpoint and output limits are available for operator changes (on maintenance level only), with a possibility for overriding the latter in the PC program. There are different ways of blocking alarms.

Apart from the supervision of the control deviation and the measured value, there is alarm or event handling for some other important errors or events as well.

Configuration

The parts of the functional unit, used in configuration, are (in analogy with other functional units):

- the PC element PIDCONA, defining some basic data and the connections to the rest of the PC program and the process
- the data base element PIDCONA (Adaptive PID control), defining data mainly for operator displays and event handling
- display elements, defining the appearance in the operator station

See Figure 1-7. The name of the data base instance ties them together.

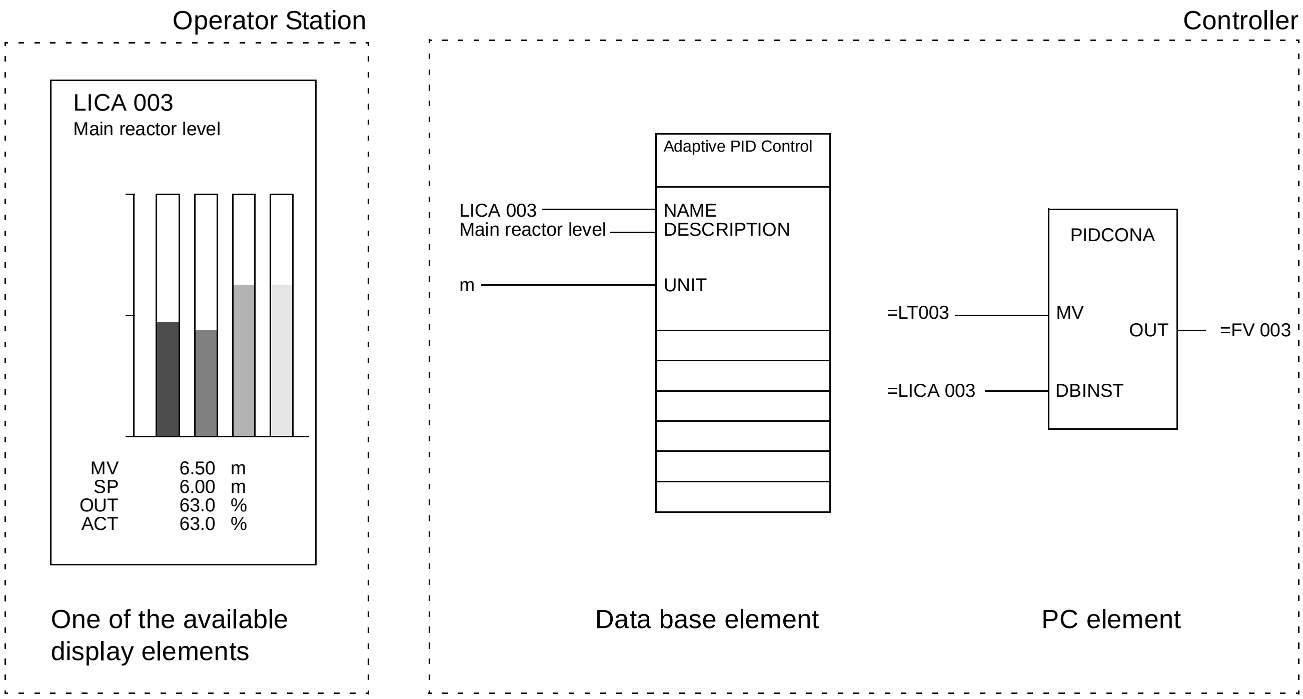


Figure 1-7. Display elements, a data base element, and a PC element (all shown very simplified here) are used in the configuration of a PIDCONA object

1.9 User Interface

All user interaction with PIDCONA is done in the standardized way used throughout the Advant Station 500 Series Operator Stations and Advant Controller 400 Series, both for building applications and for everyday operation. See *AdvaCommand Basic Functions User's Guide*, and the User's Guides for the Engineering Stations.

Chapter 2 Installation

2.1 General

No separate installation is required. The functional units are included in the control system software, ready to use when configuring for different applications.

Chapter 3 Configuration/Application Building

3.1 Design Considerations

The purpose of this section is to give useful hints regarding configuration, and to point out some design questions that should be answered, in order to get the most out of PIDCONA.

What modes to use? In many cases the modes MAN and AUTO are sufficient for the operation of the control loop. Their use is straight-forward. In other cases extended flexibility is needed, as provided in PIDCONA by the modes BAL, MAN Clamped, E1, E2, and E3. Some recommendations for their use are given in Section 3.1.2 Use of Alternative Non-Automatic Modes and Section 3.1.3 Use of Alternative Automatic Modes. It is possible to put restrictions on which modes the operator can select (using PC element inputs modeENBL), and to let some of the modes be reachable only by programmed sequences (using PC element inputs SEQmode or modeFL). The restrictions and programmed mode selections can of course be given arbitrary conditions programmed in AMPL.

3.1.1 Cycle Times

The cycle time for the PC element execution is configured in its execution unit (the control module). When selecting this cycle time you should take into account:

- process dynamics, or rather the relevant time scale of the process
- acceptable delay in responses to operator actions
- processing load
- disturbances, especially at high frequencies.

On autotuning, a sampling period for the PID control is determined automatically, adjusted to the time scale of the process. This sampling period is determined as an integer multiple of the cycle time of the control module. The configured cycle time gives a limit, the shortest possible sampling period for the control in the loop. In general, you can expect the step response settling time of the closed loop (with normal response type) to be between 5 and 50 times the obtained sampling period. If you have configured a long cycle time in relation to the process dynamics, you will get a settling time closer to the lower figure when expressed in number of samples. With a shorter cycle time, which allows a shorter sampling period for the control, the absolute settling time would be shorter, although the number of samples required for settling will be higher.

For keen operability, the cycle time 1s is normally adequate. If the processing load is very critical, you can live with a cycle time of 2s without too much frustration due to the delays.

Rule of Thumb

For slow process loops, choose the control module cycle time to be 1s (or 2s if you have load problems). For fast loops (required closed loop settling time: a few seconds or faster), choose it as fast as you need for the required performance (suggestion: 1/5 or 1/10 of the required settling time). If you do not know how fast the loop is (or should be), choose the cycle time to be 1s.

Deeper Considerations

The full executions of the PC element occur on sampling instances, in automatic mode. But the executions in between give a basic load, which directly depends on the configured cycle time. Therefore control with a sampling period of 16s, for example, would give almost four times higher load with the configured cycle time 250ms than with the configured cycle time 1s. Appendix C, PC Element provides execution time figures for your guidance.

Aliasing of disturbances in the measured value should be avoided. You should design the configuration such that there are virtually no disturbances, with frequencies above the Nyquist frequency[1] associated with the configured cycle time of the control module. If the signal from the process contains such disturbances, filtering is needed in order to reduce it to an acceptable level. For this purpose, there are filters in the analog inputs of the Controller, both hardware filters and configurable software filters.

The hardware filter of the ordinary analog input boards (bandwidth 6.5 Hz, 40 times damping at 50 Hz) suppresses disturbances picked up from mains voltage (50 - 60 Hz) effectively. If there are significant disturbances in the process in the frequency range above the Nyquist frequency mentioned, but too slow to be effectively suppressed by this hardware filter, further anti-alias filtering is needed. The signal transmitter often provides a filter that is slow enough for this purpose. If not, choose the scan time of the analog input fast enough to catch these disturbances, and use a software low pass filter to suppress them for frequencies above the Nyquist frequency associated with the configured cycle time of the control module.

It may occur that there are no disturbances to remove by filtering, or that the disturbances are effectively reduced by the hardware filter. Then there is no need to use the software filter, and no need to sample the analog input faster than the configured cycle time of the control module.

The fact that the sampling period for the control with PIDCONA is an integer multiple of the configured cycle time means that a sampling rate reduction may take place within it. There is an efficient built-in anti-alias filter, in order to avoid aliasing in this step. It is based on averaging over the whole sampling periods. If you plan to use this anti-alias with a short cycle time, to complement the hardware filter, you should be aware of that the longest possible sampling period may become too short if your process is very slow. Maximum sampling period is 2048 times the configured cycle time. For 50ms configured cycle time the maximum sampling period for control becomes 102.4s. The autotuning would require a longer sampling period, if the process is so slow that the natural oscillation of the loop with relay feedback has a period approaching an hour or longer.

3.1.2 Use of Alternative Non-Automatic Modes

The three modes BAL, MAN Clamped, and E3 are similar in one aspect, that the output copies a PC element input. There are the following differences making them fit for different purposes:

- BAL and MAN Clamped have high priority, and they cannot be selected by the operator. E3 can be selected either by the operator or by a PC element input (SEQE3 or E3FL).
- An analog input error, at the channel associated with the PIDCONA object, interrupts mode E3 and forces it to MAN, but it does not interrupt modes BAL and MAN Clamped.
- The output rate limit SPEEDO (to be compliant with the actuator's rate limit) applies to MAN Clamped and E3, but not to BAL.

---

[1] The Nyquist frequency is half the sampling frequency

Recommendations for their use:

- Use BAL when the relevant value for the output is generated and fed to the actuator without any control from the PIDCONA object, or in situations where there is no actual "actuator" and just a value "as good as possible" has to be used as starting level for the output. Typical case: The outer loop in a cascade should be in mode BAL, when the inner loop is in mode MAN. This is exemplified in Section 3.1.3 Use of Alternative Automatic Modes and Figure 3-3.
- Use MAN Clamped when the output shall be forced to a predetermined value for some critical situation.
- Use E3 for an alternative controller used in place of PIDCONA, or when the output is generated in some other way. Especially if the same measured value is used, if the operator shall be able to select the mode, or if the logics selecting it belong to the PIDCONA object.

Figure 3-1 shows a configuration example with an alternative controller used in mode E3. In this example, the user interface for PIDCONA is also used for selecting the mode of the alternative controller (auto if PIDCONA is in E3, and otherwise manual), and for entering its setpoint and output limits. Setpoint tracking in PIDCONA is used in modes MAN and BAL, but not in mode E3. Bumpless mode transfers are ensured, including switching between the two controllers, by letting them know of each other's output values. Autotuning in mode E3 has been blocked, since it would confuse the alternative controller.

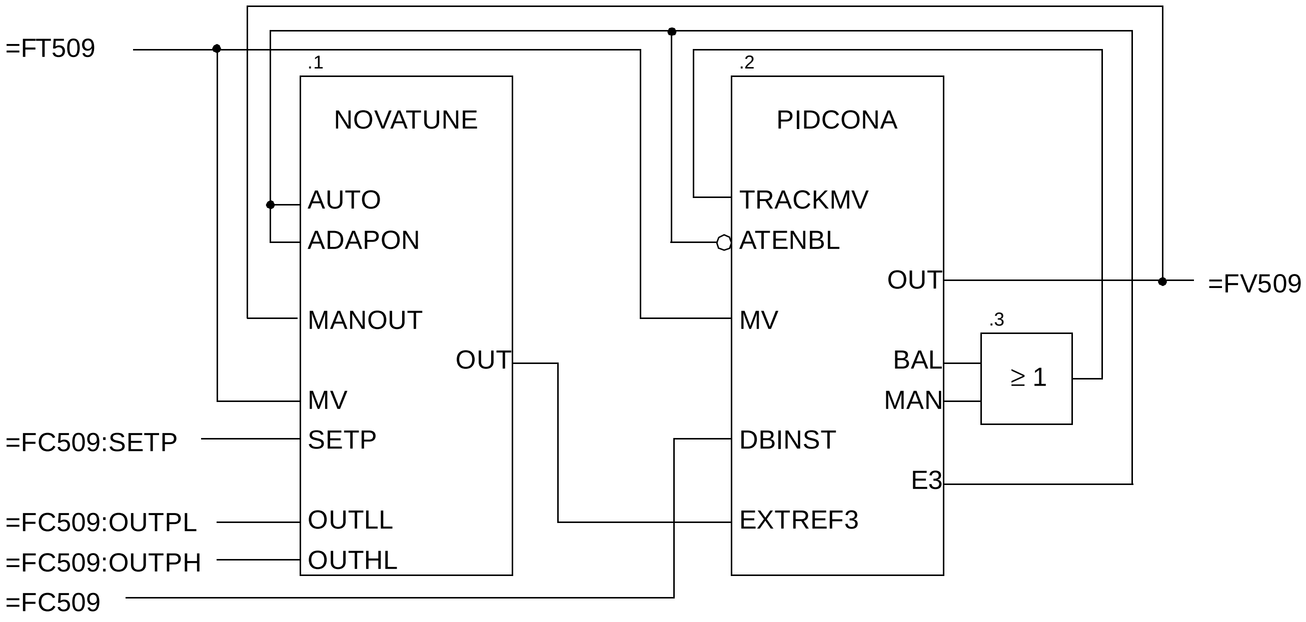


Figure 3-1. Configuration example using an alternative controller (NOVATUNE) in mode E3. Here the operator interface of PIDCONA is always used in the control loop, i. e. for the alternative controller too.

3.1.3 Use of Alternative Automatic Modes

The modes E1 and E2 provide automatic control using externally supplied setpoints. They are totally equivalent to each others. A cascade control is shown in Figure 3-2. This is a typical case where mode E1 or E2 will be used.

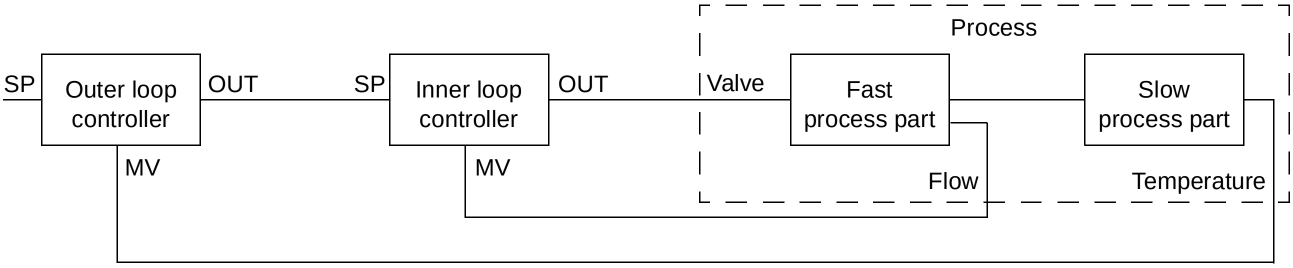


Figure 3-2. Cascade control. Valve, Flow and Temperature indicate an example of what kind of process it can be used for.

A suitable configuration using PIDCONA is shown in Figure 3-3, where mode E1 of the inner loop is used for cascade control. By choosing BUMPLESS=1 or TRACKMV=1 (or both) it is ensured that WSP tracks the measured value in non-automatic modes, in such a way that it is suitable for the balancing of the outer loop, in all modes where the cascade control is not in use. By choosing TRACKEXT=1 or 2, the operator's setpoint is updated with the external setpoint values, making the transfer from E1 to AUTO free of change.

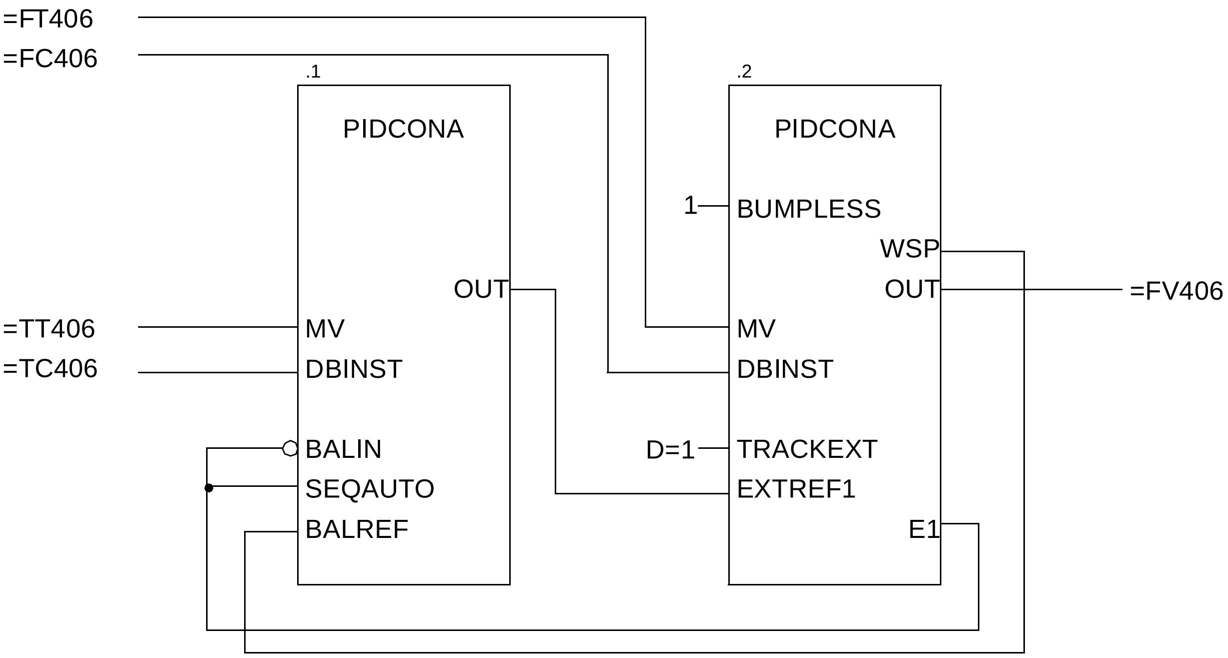


Figure 3-3. Configuration example for cascade control with two controllers of type PIDCONA. Cascade control is obtained when FC406 is in mode E1 and TC406 is in mode AUTO.

Mode switching is in this example assumed to be performed with the dialog of the inner loop. Commanding mode E1 automatically sets the outer loop in mode AUTO, and in all other modes of the inner loop, the outer loop is balancing. You can still, while the inner loop is in mode E1, put the outer loop in mode MAN. This is functionally equivalent to having the inner loop in mode AUTO, except that the inner loop setpoint will be manipulated using the OUT dialog of the outer loop.To avoid this, you can omit the possibility by connecting E1 to AUTOFL instead of to SEQAUTO. Then operator commands for mode MAN have no effect since they have lower priority than the programmed command for mode AUTO. Equal function is obtained if MANENBL=0 in the outer loop. If E1 of the inner loop is not connected to any of SEQAUTO or AUTOFL of the outer loop, the outer loop will go to mode MAN when the inner loop goes to mode E1. It is then left to the operator to activate the cascade control, by selecting the outer loop and switching its mode to AUTO.

There is another comment about the example in Figure 3-3. When the inner loop output in a cascade hits its limits, the configuration should prevent the outer loop from further changes of its output in that direction. If the outer loop runs adaptive control this becomes even more important. Consider to turn off its adaptation during such periods. Figure 3-4 shows a suggested solution (Swap the connections to OUT=LL and OUT=HL if you have REVACT=0.)

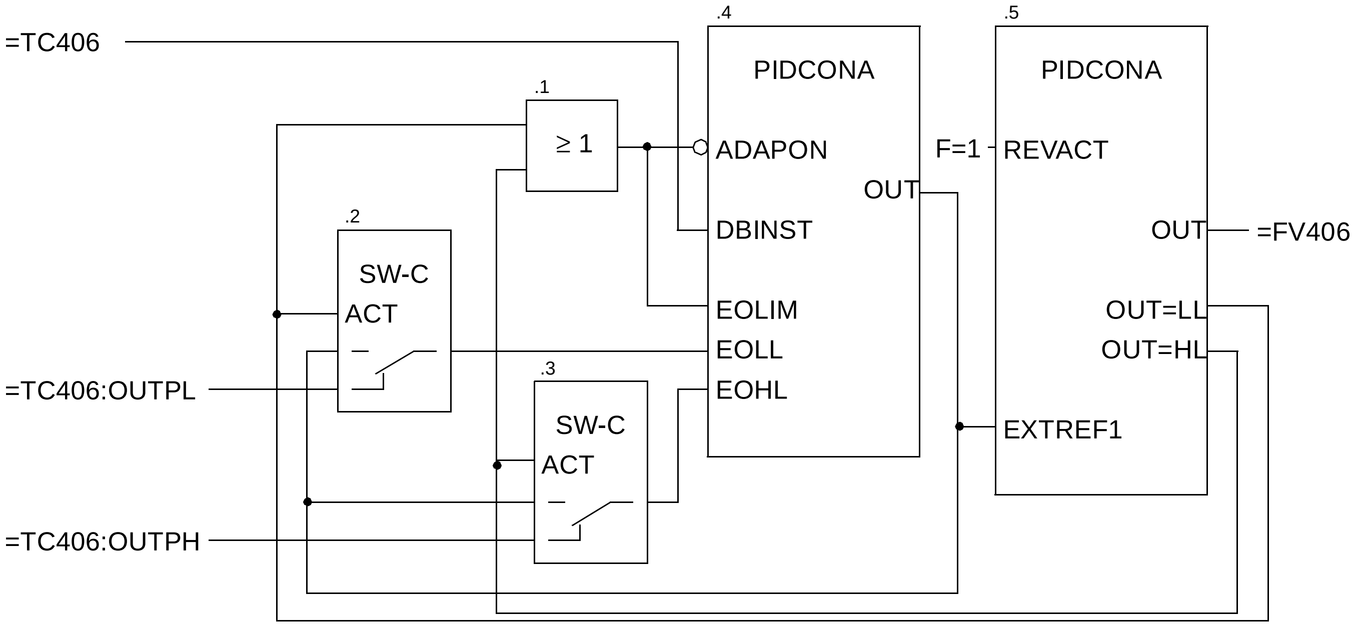


Figure 3-4. Example showing a way of configuring the outer loop in a cascade to limit its output and switch off the adaptation when the inner loop hits a limit

In a cascade, the inner loop is often much faster than the outer one. A cycle time chosen for adequate response time for the operator such as 1s or 2s, may be too slow for the inner loop, but still fast enough for the outer loop. Then, to reduce the processing load, you should place the two controllers in different control modules with different cycle time, rather than running both at the faster rate needed for the inner loop.

Other typical uses of modes E1 and E2, beside cascade control, are when the setpoint values are provided by a recipe, or when they are obtained by an optimizing program.

3.1.4 Reverse Action or Direct Action

The direction of the controller's action is configured with the PC element functional parameter REVACT. If the final response of the process (the measured value) to a change of the controller output is in the same direction, the controller shall operate with reverse action, REVACT=1. This means that a deviation in one direction is counteracted with an output change in the opposite direction. The opposite is direct action, REVACT=0, when a deviation in one direction is counteracted with an output change in the same direction. Direct action is used if the final response of the process to a change of the controller output is in the opposite direction

3.1.5 Output Rate Limit and Hard Limits

If the output goes directly to an actuator, you can probably find out how fast it is able to move. Then enter this value as the rate limit SPEEDO to ensure that the controller's knowledge of the position is relevant. The actuator may have hard limits inside the output range (defined by the data base element terminals OUTMIN and OUTMAX). Then make sure, also, that the output limits for PIDCONA are not outside these hard limits.

The reason why this is important is that so called reset wind-up may occur, if the controller works with fictitious output values outside what the actuator can reach. A discrepancy between controller output and actual actuator position causes even more serious problems if the control is adaptive, since it will then corrupt the tuning. This holds also for the autotuning.

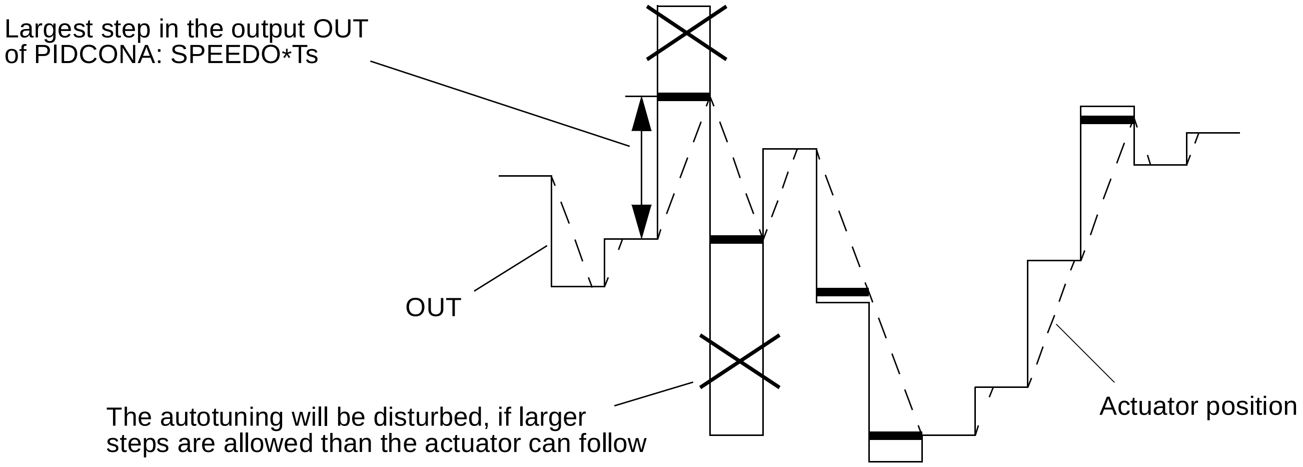

Largest step in the output OUT
of PIDCONA: SPEEDO*Ts
OUT
The autotuning will be disturbed, if larger
steps are allowed than the actuator can follow
Actuator position

Figure 3-5. Select a suitable value of SPEEDO with respect to how fast the actuator can move

Also in cases where the output is not directly connected to an actuator, there may still be relevant physical limitations, which ought to be taken into account in a similar manner. For example, the outer loop in a cascade should (by a suitable choice of SPEEDO) avoid moving its output so quickly that the inner loop hits its rate limit.

3.1.6 Actuator Position and Direction

The actuator position can be shown in some of the display elements (group display, object close-up display, and object display). It requires both that SHOW_ACT is set, and that the value to show is connected to ACTPOS, in the data base element. The scale and the unit are the same as for the output. The data base element terminal ACT_DIR determines whether 100% output will be indicated as open (Op) or closed (Cl).

3.1.7 Scaling and Units

The range for the measured value and the output and their units can be chosen freely to suit the process. The gain presented in the object display, and available in the data base elements, is expressed for the scaled units (percents), where the specified ranges correspond to 0…100%. The range for the scheduling input is also specified. It affects the range presented in the object display, and also the minimum hysteresis at zone limits, which is one percent of the range.

3.1.8 Gain Scheduling versus Adaptive Control

Gain scheduling has the advantage over truly adaptive control, that it is very quick in changing the settings. It does not have to wait for an identification to detect the change of the process dynamics. On the other hand, it requires knowledge of a scheduling variable that has some relation to the changes of the dynamics, enough to indicate with repeatability which setting to use from the schedule.

3.1.9 Precautions in Adaptive Control

A basic rule is that you should not use adaptive control unnecessarily. When you do use it, make sure that adaptation does not take place when the conditions for adapting correctly are not at hand. If the process is not operating normally, you do in most cases not want the controller to adapt to those circumstances. If it has been operating under such abnormal conditions, with adaptive control, and comes back to a normal state, you would like it to return to a normal setting quickly.

Make use of the PC element inputs ADAPON and RESTORE. Reset ADAPON while it is not suitable to adapt. Apply a positive transition to RESTORE, when it is better to restore the saved setting (and possibly continue adapting from it), than to continue adapting from the present one.

3.1.10 Feedforward

Disturbances that are measurable and have a known effect on the process should be taken care of by feedforward. PIDCONA has the PC element input ADDOUT for this purpose. Implement in AMPL a calculated correction based on the measured disturbances, and connect it to ADDOUT. It will then be added to the output of the control algorithm, and the result will be checked against limits and rate limits, thereby giving a correct back calculation when needed. Feedforward is most effective for fast disturbances, and changes of the ADDOUT value affects the output at every PC element execution, irrespective of the sampling period for the feedback control. The feedforward is also active during autotuning, to ensure that the tuning is done under conditions similar to normal operation.

3.1.11 Output Reference Level in Proportional Control

When used without integral action, you may want to specify what level the output of the controller shall have at zero deviation. The control algorithm itself will deliver zero in this situation, and then the value of the PC element input ADDOUT is added. With a constant value it acts as reference level. If, for example, you want the output to vary around the middle of its range for deviations around zero, you set ADDOUT equal to this middle value (50 if the range is 0…100). This is valid for P and PD control. As soon as integral action is included, which it will always be when the controller is automatically tuned, the corresponding level is found automatically.

3.1.12 Derivative Part Choice

The design method, used during autotuning or adaptive control, chooses the BETA value to give a good setpoint response. It is then assumed that the D part acts on the measured value, i. e. the PC element's functional parameter DERIVDEV is zero. Thus, it is recommended as preferred choice to have DERIVDEV=0. In spite of that, it is possible to choose DERIVDEV=1, which means that the D part acts on the deviation DEV instead. That choice can be expected to give more overshoot on setpoint changes.

3.1.13 Programmed Blocking of Limit Alarms

The PC element input ALCBLK, when set, blocks the limit supervision within the PC element. The limit status outputs are reset (MV<L2, MV<L1, MV>H1, MV>H2, DEV<L, and DEV>H). The blocking is not indicated in display elements.

Signals connected to the data base element terminals ABL_H1L1, and ABL_H2L2 block alarms without affecting the PC element status outputs. It is indicated in display elements that the alarm is blocked.

Consequently, if the condition shall block the limit supervision completely and it shall be indicated in the operator displays, then connect the blocking signal both to the PC element input ALCBLK and to the data base element terminals ABL_H1L1 and ABL_H2L2.

3.1.14 Alternative Event Treatment

The default values in the event treat part of the data base element give a consistent treatment of events and alarms in agreement with other functional units. You can customize the treatment using other values of the event treat pointers, and you can form alarm groups. Please refer to *AdvaCommand Basic Functions User's Guide*, which contains a complete description of how to do this. Its list of standard event texts is also reproduced in Appendix E, Standard Event Texts. See also the "Status List" function, described in *AdvaCommand Status List User's Guide*.

3.1.15 Deviation in Non-Automatic Modes

The deviation is supervised against high and low limits, unless these alarms are blocked. This is done either if there is feedback control or not. The deviation is the difference between the measured value and the setpoint. When using automatically tuned control, the deviation supervised is the difference between the measured value and an expected value calculated from the setpoint. One could question what value to use while there is no feedback control. This is cleared out below, thereby explaining how the deviation alarm works in non-automatic modes.

Although it is not active, there is a value present for the 'working setpoint', WSP, also in the non-automatic modes, BAL, MAN and E3. It is available as a PC element output, presented as SP in operator displays, and used in the calculation of the deviation, DEV. The value of WSP, and consequently also DEV, in non-automatic modes depend on the choice of tracking and bumpless options. (For a detailed description of the working setpoint, see Section 4.1.4, Setpoint Generation, Ramping, and Tracking.)

If you have specified bumpless transfer (BUMPLESS=1) and/or to track the measured value (TRACKMV=1), the working setpoint, WSP, is balanced against the measured value (Anti-Alias Filtered), MVAAF, in modes BAL, MAN, and E3. The deviation becomes zero (except in no integral action where a correction of WSP appears in the deviation).

If you have specified neither bumpless transfer nor to track the measured value (i. e. you have BUMPLESS=0 and TRACKMV=0), then the working setpoint, WSP, equals the operator supplied setpoint, AUTOSP, in modes BAL, MAN, and E3. The deviation will then be the difference between the measured value, MVAAF, and the setpoint, AUTOSP.

In the latter case the deviation may easily become large. Depending on the application, this may or may not be a relevant deviation to supervise. Take this into account, when you decide how to use the deviation alarm, and how to choose the tracking and bumpless options. The example of Figure 3-1, shows a configuration where (assuming BUMPLESS=0, at least in mode E3) the deviation supervision is used correctly in mode E3, and deviation alarm will not occur in modes BAL and MAN, since it has TRACKMV=1 in these modes.

3.2 Capacity and Performance

The execution time and the memory required for the PC element is given in Appendix C, PC Element. In addition to those figures, 2240 bytes are allocated per instance for use by the PC element and 4908 bytes per instance for the adaptation task. The number of instances assumed for this allocation is given by the data base dimensioning (DIMDB). There is also required, per created instance, 1130 bytes of memory for the data base. Due to the physical file size in the data base there is a theoretical upper limit of 203 PIDCONA instances in the same Controller.

The adaptation involved in automatic tuning and adaptive control is performed as a background task in the Controller. With slow sampling rates or little load in the Controller, the adaptation calculations will be performed for each sample. With higher rates or higher load, there is not enough processing capacity to allow this. Some samples are then skipped, and the only effect is that adaptation takes a bit longer time, than if all samples had been treated.

If several control loops run adaptive control or autotuning at the same time, they are all served by the same adaptation task. All loops are guaranteed to be served. A fast loop cannot block the adaptation of a slow loop. If there is not processing capacity to treat all samples of all loops, some samples will be skipped, again with the result that the adaptation will be slower.

3.3 Application Start-up

For a description of the start-up of the Controller you are referred to its User's Guide. The behavior of the functional unit PIDCONA at start-up and different types of restart is described in *Functional Units Part 1 Summary and Common Properties*.

The memory allocation mentioned in Section 3.2 Capacity and Performance takes place when the Controller is taken out of configuration mode (DICONFIG command) for the first time after a cold start. In case there is not enough memory left in the Controller for this operation, a system message is issued, and the PC element will not execute its code. If this has happened, the mode indications for PIDCONA instances in this Controller will not show up, neither in the operator displays nor in the PC element outputs.

3.4 Tutorial

A control loop using PIDCONA is configured in the way described in *Functional Units Part 1 Summary and Common Properties*. The procedure is described in Section 3.5 Application Procedures. The first example below is an illustration of the steps for a straight-forward case. A second example illustrates the use of gain scheduling.

3.4.1 Basic Configuration Example

The level of a reactor tank is going to be controlled using the valve position in the inlet as manipulated variable, Figure 3-6. The valve actuator is connected to the analog output tagged LV003 of the Controller. The direction is such that increasing the output value means opening the valve. The level measurement is available as analog input tagged LT003. Its range is 0…12 m. The loop is described as ‘Main reactor level’, and it is given the tag name LICA003. .

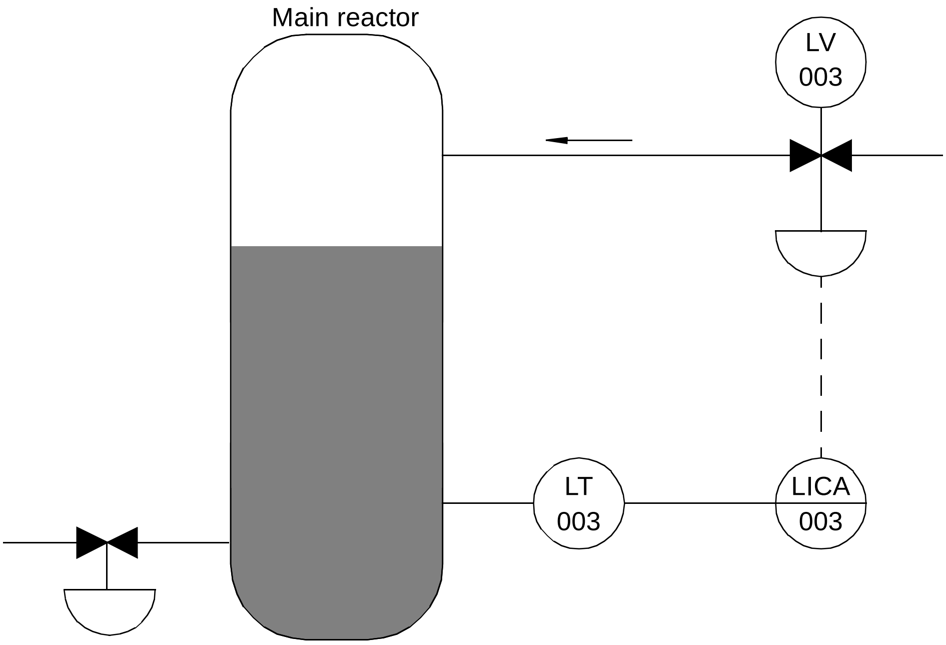


Figure 3-6. The level control to implement

The Data Base in the Controller

First we create a data base instance of PIDCONA and fill in some data in Base Part 1. The default values in Base Part 2 are okay for our application and are not changed. Figure 3-7 shows the data base element with Base Part 1 expanded, after the data has been entered.

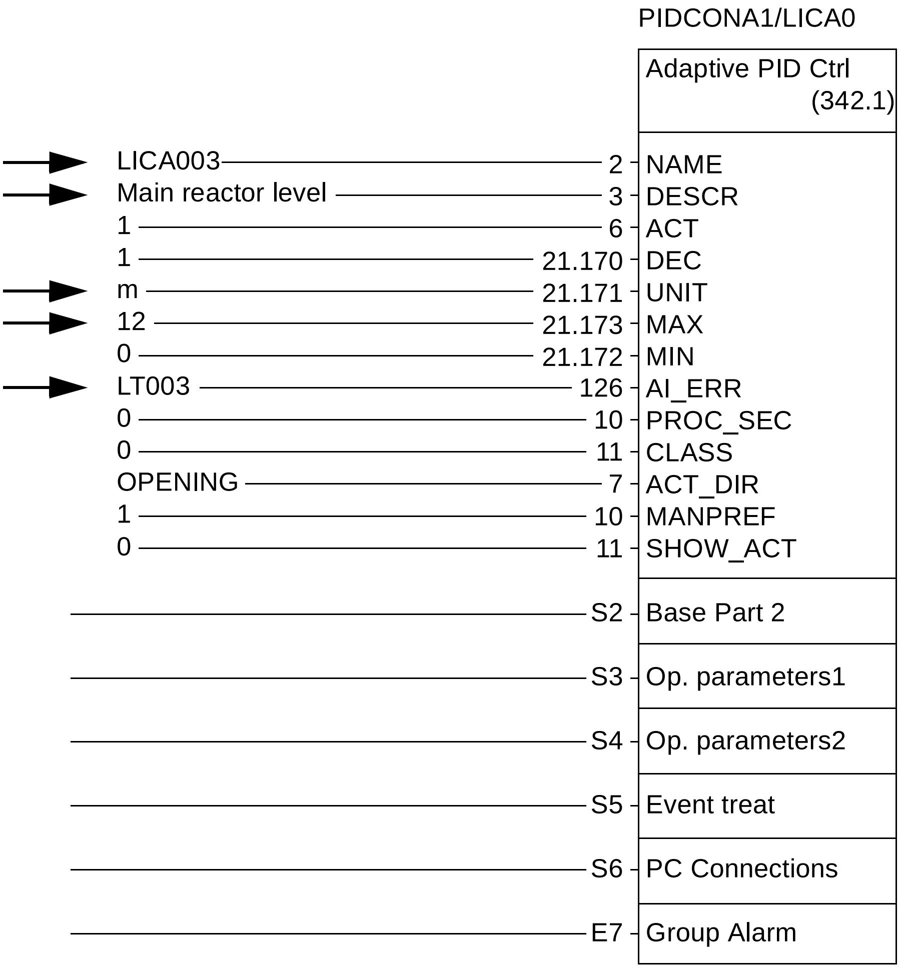


Figure 3-7. The PIDCONA data base element after entering data for the reactor level control example (only those indicated by an arrow were changed from their default values)

The PC Program

When designing the PC program, one of the things to decide is what extensions to include in the PIDCONA PC element. In this example, external reference modes will not be used, and not the mode BAL, or clamping, or external output limits either. We do therefore omit the extensions C1, C2, C3, and C5. We do not have any reason to believe that the gain or time constant of the process varies, so gain scheduling will not be used. The extension C6 is thus not included either. The only extension we include is C4, in order to get limit supervision of the reactor level and of its deviation from the setpoint. With no interlocking or other conditions to take into account, the PC program then becomes as simple as shown in Figure 3-8.

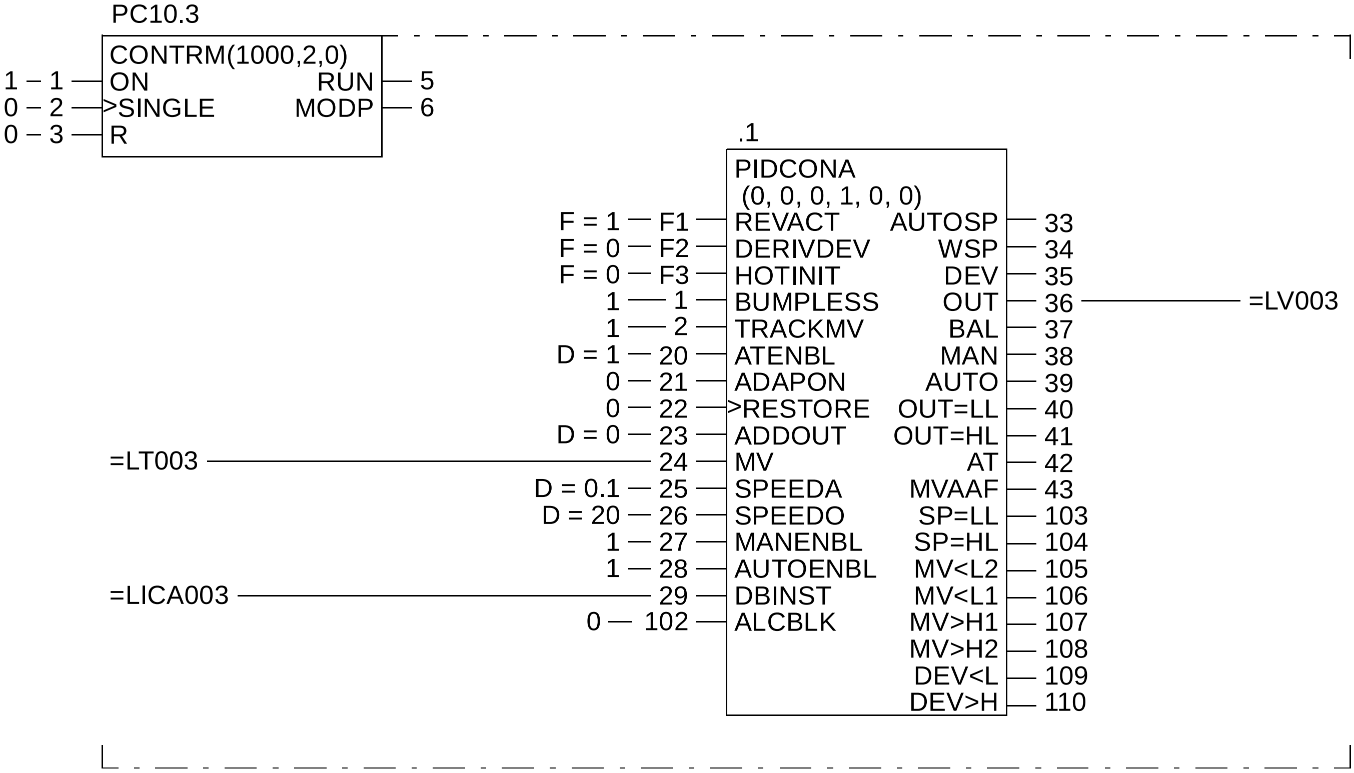


Figure 3-8. The PC program for the reactor level control example

The process is rather slow, and we have chosen the cycle time of the control module to be one second. We have also made the judgement that a suitable ramping speed for the setpoint, when the operator changes it in a larger step, would be 6 meters per minute, i. e. 0.1 meter per second. This was entered for SPEEDA. Before entering a value for SPEEDO, the output rate limit, we have checked the data sheet for the actuator. We found that, at its maximal speed, it moves between fully open (100%) and fully closed (0%) in five seconds. This we translate to a maximum speed of 20 percents per second, which we enter for SPEEDO.

We have reverse action (REVACT = 1) since a too low level shall be counteracted by opening the inlet valve more, i. e. by increasing the output from the controller. (Since a change of the controller output influences, in stationarity, the measured value in the same direction, a deviation in one direction is to be counteracted by an output movement in the reverse direction.)

The operator setpoint is chosen to track the measured value (TRACKMV=1) in mode MAN. If needed, the operator will adjust the setpoint value after changing mode from MAN to AUTO, and not before. The value of BUMPLESS has no influence in this case, when we have TRACKMV=1 and do not use modes E1 and E2.

Since all mode changes are done by the operator we have set MANENBL and AUTOENBL to one permanently.

We will not use adaptive control, and therefore we have put ADAPON (and RESTORE) to zero permanently. Autotuning will be used, but we may choose to reset ATENBL afterwards, which prevents anyone from doing a new autotuning without first setting ATENBL again.

We have no signal to use for feedforward, and we will have integral action in the controller (it always gets that when autotuned). Therefore we have ADDOUT=0.

We do not use programmed alarm blocking. ALCBLK is zero permanently.

Displays

Now that the data base instance is created and the PC program is entered in the Controller, it is possible to run the control loop. But until the object has been included in any process displays, the only way to access it in the operator station is via an object request by name. That way we can bring up the dialog and the object display directly.

To get an overview of the process, and a convenient way of selecting the loop for manipulations, we build a process display, using the display elements available for the PIDCONA type of object together with other display elements. See Figure 3-9.

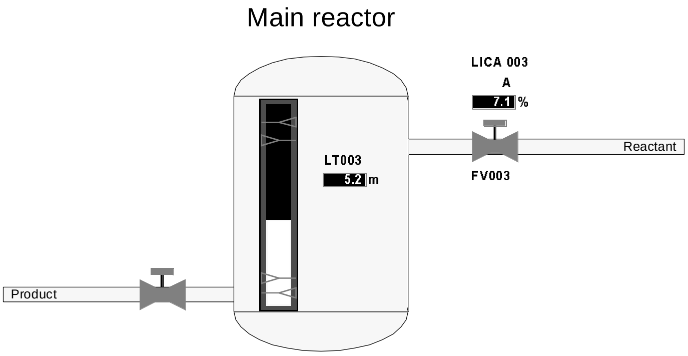


Figure 3-9. A process display part, built for the reactor level control example.

3.4.2 Configuration example with Gain Scheduling

This example presents a case with gain scheduling. It is less detailed concerning standard configuration questions than the example of Section 3.4.1 Basic Configuration Example.

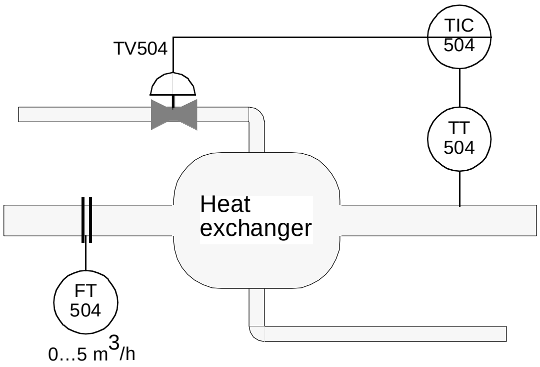


Figure 3-10. Temperature control of a heat exchanger

We have a heat exchanger using steam to heat our process medium. There is a steam valve, TV504, a temperature measurement TT504 in the heated process medium, and a flow measurement, FT504, as shown in Figure 3-10. (It is a case without big steam pressure variations, so a local steam flow loop is not needed.) We will now configure the temperature control loop, TIC504.

In a heat exchanger the flow of the heated medium has great influence on both the deadtime and the process gain in the temperature control loop. They are both roughly proportional to the inverse of the flow. Therefore, it is natural to apply gain scheduling with the flow as scheduling input.

In this example the range for the flow is 0…5 m^3/h. Thus, SCHMAX, in the data base element, Base part 2, is set to 5. The temperature range, the supervised AI channel, and the name and description are defined in Base part 1. The data base element with these values entered is shown in Figure 3-11, where Base part 1 and Base part 2 are expanded (the only parts concerned by the configuring).

In the PC element, the flow measurement is connected to the input SCHEDIN. Gain scheduling is enabled with the input GSENBL. The zone limit inputs ZLIM12, … ZLIM45 define a scheme with narrower zones for low values of the flow. This choice is motivated by the knowledge that flow variations have greater influence on the process for low flow values. The setpoint rate limit SPEEDA is chosen to give a ramp speed of 15 degrees per minute (0.25 degrees per second). The output rate limit SPEEDO=5 corresponds to an actuator needing 20 seconds for a full

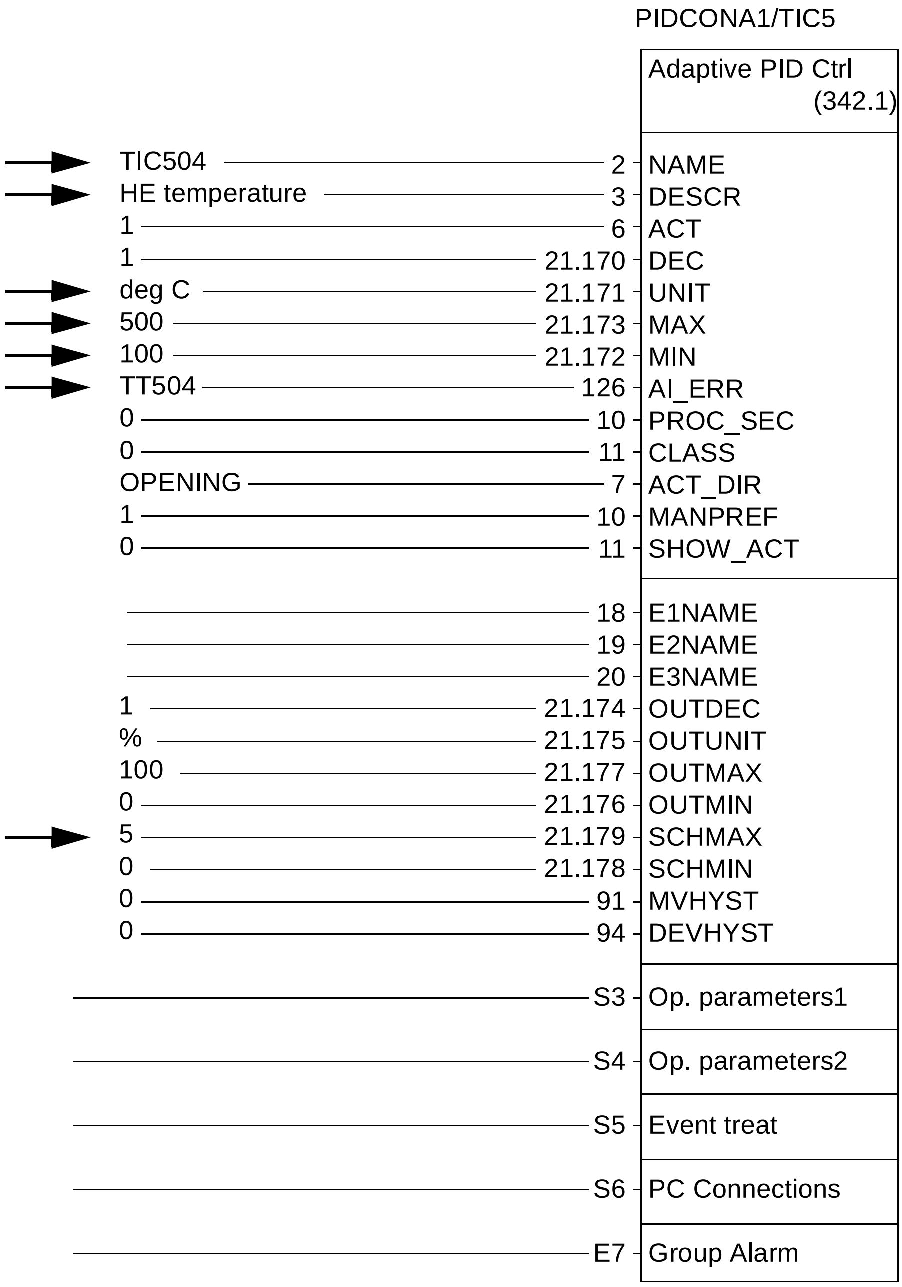


Figure 3-11. The PIDCONA data base element after entering data for the heat exchanger control example (only those indicated by an arrow were changed from their default values)

movement between 0% and 100%. Having the remaining inputs based on the same considerations as in the previous example, the PC solution is presented in Figure 3-12.

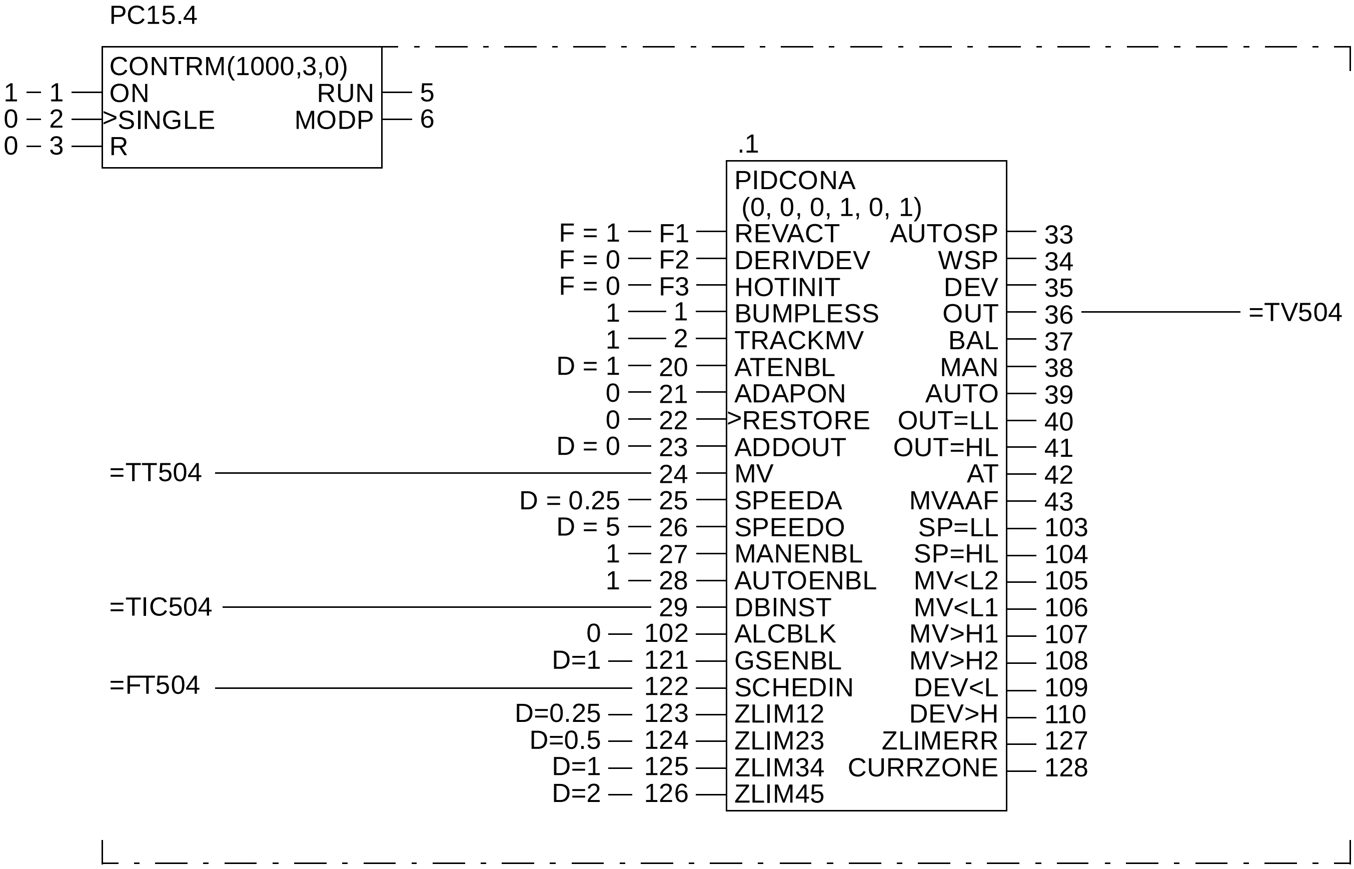


Figure 3-12. The PC program for the heat exchanger temperature control example

3.5 Application Procedures

The general procedure for configuration of a control loop is described in *Section 2.3 of Functional Units Part 1 Summary and Common Properties*.

In the Controller, you create a data base instance of PIDCONA, and you fill in the name and description. The other properties in Base Part 1 and Base Part 2 of the data base element have default values, and you may need to change some of them. This way you define (default being given in parenthesis)

- the range and unit used, and the number of decimals shown in the operator displays, for the measured value. (0…100 %, one decimal)
- the range and unit used, and the number of decimals shown in the operator displays, for the output. (0…100 %, one decimal)
- the range for the gain scheduling input. (0…100)

- the channel for analog input error supervision. (None)
- the process section and class, to use for alarms and status lists. (0, 0)
- the hystereses for limit alarms. (0)
- whether 100% output shall be shown as Open or Closed. (Open)
- whether the actuator position shall be shown or not. (Not shown)
- optional names of the external reference modes, E1, E2, and E3. (*"empty string"*)

If you are going to use non-standard event handling, you may change some properties in the Event Treat part of the data base element as well. The rest of the data base properties are left unchanged. They will normally be updated and used in run-time operation.

When you create the instance, you get both a superior data base element and a subordinate one. You usually do not have any need to access the subordinate one. If required, you can access it from the superior element, with the command '>>'. The subordinate element holds data regarding saved settings and models, this way being retained when an application is moved using data base source code. Appendix D, Data Base Elements describes the two elements.

A PC program defines the control function performed in the Controller. For the design of the PC program there are hints and advice in Section 3.1 Design Considerations. See also Appendix C, PC Element and the detailed description in Section 4.1, Product Operation of how PIDCONA operates in runtime.

When you have decided what functions in PIDCONA will be used, you can conclude which of the extensions C1, …, C6 should be included in the PC element. Once you have entered the PC program into the Controller, you cannot change what extensions are included except by deleting and reinserting the PC element.

Function required	Include extension
Modes E1, E2, and/or E3	C1
Programmed selection of modes E1, E2, and/or E3	C2
Mode BAL	C3
Programmed selection of modes MAN and/or AUTO	C3
Supervision of measured value and deviation against limits	C4
Clamping of the output (mode MAN Clamped)	C5
Externally supplied output limits (programmed limits)	C5
Gain scheduling	C6

Since the PC element connection DBINST usually references the data base instance by name, the instance should be created before you enter the PC program.

In the Operator Station, you use the display elements in composing process displays, according to methods described in *AdvaBuild Display Builder User's Guide*. The object display and the standard alarm handling are available without any configuring. To use the object trend, you need to create and parameterize a TTDLOG data base instance in the Controller, and then to configure the appropriate logs in the Operator Station, as described in *AdvaCommand Basic Functions User’s Guide*. The last mentioned manual also describes how you configure events and alarms in case you want to customize, and how to configure group alarm. For alarm sectioning we refer to *AdvaCommand Process Sectioning*. Another configurable function applicable to PIDCONA is the status list, described in *AdvaCommand Status List User’s Guide.*

3.6 Configuration/Application Building Menus

Consult the User’s Guides for your Engineering Station and your Operator Station.

Chapter 4 Runtime Operation

4.1 Product Operation

This section gives a detailed description of the function of PIDCONA.

4.1.1 Control Modes

The manual *Functional Units Part 1* gives a general description of the control modes, BAL, MAN, AUTO, E1, E2, and E3. Their exact function and the conditions governing their choice in PIDCONA are described below. The control modes of PIDCONA are listed in Table 4-1 and Table 4-2. During normal operation, the control algorithm (the feedback) is active only in the automatic modes. It is different during an autotuning session, as outlined in Section 4.1.7 Autotuning.

Table 4-1. Automatic modes

MODE	Source of Setpoint
AUTO	AUTOSP, setpoint value entered by operator
E1	EXTREF1, external reference number 1 (PC element input)
E2	EXTREF2, external reference number 2 (PC element input)

Table 4-2. Non-automatic modes

MODE	Source of Controller Output
BAL BAL Local (1)	BALREF (PC element input)
MAN Clamped (1)	CLAMPREF (PC element input)
MAN MAN Forced (1)	MANOUT, the operator supplied manual output value
E3	EXTREF3, external reference number 3 (PC element input)

(1) The modes BAL and MAN can have the additional statuses shown in the table.

In automatic modes, the output of the controller (the PC element) is generated by the control algorithm, with the value of the PC element input ADDOUT added. The generation of the output is further described in Section 4.1.5 Output Generation and Ramping.

The modes E1, E2, and E3 are available only with the PC element extension C1 included.

Table 4-3. Control mode selection in normal operation. Command signals are listed in priority order. The shading indicates the mode selected, and its additional status where appropriate.

Command		**Mode selected**					
signal	**source**	**BAL**	**MAN**	**AUTO**	**E1**	**E2**	**E3**
LOCALFL (1)	PC	Local					
LOCAL (1)	PC	Local					
BALIN (1)	PC	▨					
CLAMP (1)	PC		Clamped				
Man Forced (2)	Operator		Forced				
AI error alarm	AI channel		▨				
MANFL	PC		▨				
AUTOFL	PC			▨			
E1FL	PC				▨		
E2FL	PC					▨	
E3FL	PC						▨
Man	Operator		▨				
SEQMAN ↑ (3)	PC		▨				
Auto	Operator			▨			
SEQAUTO ↑ (3)	PC			▨			
E1	Operator				▨		
E2	Operator					▨	
E3	Operator						▨
SEQE1 ↑ (3)	PC				▨		
SEQE2 ↑ (3)	PC					▨	
SEQE3 ↑ (3)	PC						▨

(1) On leaving mode BAL or MAN Clamped (i. e. when the command signal returns to zero), mode MAN is always obtained, unless any signal commands another mode.

(2) The mode MAN Forced is temporarily left while a command signal with higher priority is active. The mode is not left finally, until the operator gives a command for another mode. All other commands of lower priority are blocked, but any active "modeFL" signal will have effect as soon as MAN Forced is left.

(3) The arrow ↑ indicates that only the positive transition of the signal is relevant for commanding a mode change.

In Table 4-3 the selection of control mode is illustrated. Normally a command signal with lower priority has effect if no higher priority command signal is active, and only then. As long as no command signal is active, the mode stays unchanged. The exceptions from these two rules are stated in the footnotes to the table. The operator commands can be enabled or blocked by the PC element inputs MANENBL, AUTOENBL, E1ENBL, E2ENBL, and E3ENBL respectively. The PC element input command signals are available in appropriate extensions of the PC element.

During autotuning, the selection of control mode is restricted. Some of the command signals abort the autotuning and some are ignored. The selection is shown in Table 4-4. During the autotuning, the mode will determine which setpoint value to use. Whether the control algorithm is applied or not is determined entirely by the autotuning phases. The mode selected will be applied when the autotuning is left (unless aborted the way indicated in the table). After that the mode is determined according to the conditions of Table 4-3.

Table 4-4. Control mode selection during autotuning. Command signals are listed in priority order. The shading indicates the mode selected, or whether the autotuning is aborted.

Command		**Aborts autotuning**	**Mode selected**				
signal	**source**		**MAN**	**AUTO**	**E1**	**E2**	**E3**
LOCALFL (1)	PC	▨					
LOCAL (1)	PC	▨					
BALIN (1)	PC	▨					
CLAMP (1)	PC	▨					
Man Forced (1)	Operator	▨					
AI error alarm (1)	AI channel	▨					
Man	Operator		▨				
Auto	Operator			▨			
E1	Operator				▨		
E2	Operator					▨	
E3	Operator						▨

(1) If the autotuning is aborted by any of these command signals, the mode is determined as shown in Table 4-3.

4.1.2 Control Algorithm

The control applied in automatic modes (AUTO, E1, E2), is of type PID (Proportional, Integral, and Derivative action). The PID control is essentially described by the formula

$$u = \mathrm{K}\left[\beta r - y + \frac{1}{\mathrm{T_I}}\int(r - y)dt - \mathrm{T_D}\frac{dy}{dt}\right]$$

but of course a sampled version is used. When required, filtering is automatically applied to the derivative part. The algorithm is also illustrated by the block scheme in Figure 4-1.

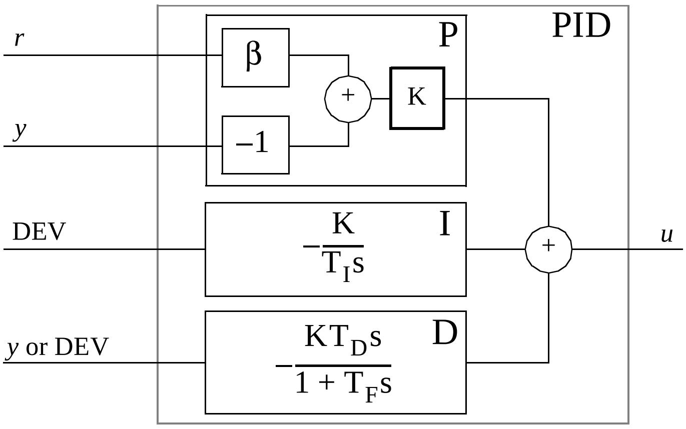

r
y
DEV
y or DEV
PID
P
β
−1
+
K
I
$-\frac{\mathrm{K}}{\mathrm{T_I s}}$
D
$-\frac{\mathrm{KT_D s}}{1 + \mathrm{T_F s}}$
+
u

Figure 4-1. The control algorithm

The notations used are:

u is the output from the control algorithm

y is the measured value (anti-alias filtered, i. e. MVAAF), and

r is the working setpoint, WSP.

All (u, y, r) are expressed in percents. The deviation DEV is y - r.

K is the gain with sign, i. e. K = GAIN if REVACT=1 (reverse action) and K = – GAIN if REVACT=0 (direct action). Here GAIN is the present value in the data base also available in the object display, and REVACT is a PC element function parameter.

$\mathrm{T_I}$, $\mathrm{T_D}$ is the integration time and derivative time respectively, both given in seconds which is also the unit for the integration and the derivation in the formula.

β is the setpoint factor.

$\mathrm{T_I}$, $\mathrm{T_D}$, and β are the present values of TI, TD and BETA in the data base, also available in the object display.

$\mathrm{T_F}$ is the filter time constant in the D part. It is not available to the user. It is chosen automatically when needed to keep the high frequency gain below ten times the mid-frequency gain.

When TD is zero there is no derivative action. Then the action is either P or PI.

If TI is zero, there is no integral action. (It is interpreted that way although the formula above would imply infinite integral action with $T_I = 0$.) Then the action is either P or PD.

The setpoint factor β appearing in the P part makes the formula differ from the most common form of a PID controller (in which case this factor has the fixed value = 1, the P part thus operating on the deviation DEV). The inclusion of β allows the loop to be made faster without causing big overshoots at setpoint changes. To get an ordinary P or PD controller without a reducing setpoint factor, you have to choose BETA = 1.

The design method, used during autotuning or adaptive control, chooses the BETA value to give a good setpoint response. It is then assumed that the D part acts on the measured value, i. e. the PC element's functional parameter DERIVDEV=0. In spite of that, it is possible to choose DERIVDEV=1, which means that the deviation DEV replaces the measured value y in the D part. Choosing DERIVDEV=1 would be expected to give more overshoot on setpoint changes.

The output u of the control algorithm is converted from percents to physical units. To this value the PC element input ADDOUT is added. The result is checked against high and low limits and rate limit, as described in Section 4.1.5 Output Generation and Ramping. Then it is fed to the PC element output OUT.

There is an Anti Reset Wind-up function which prevents the integrating part from winding up in cases when the controller output hits the limits. This is done with a kind of back calculation, using the values actually obtained after the output ramping unit (described in Section 4.1.5 Output Generation and Ramping). It also takes into account the values of ADDOUT.

When going from a non-automatic to an automatic mode, the control algorithm is always initialized to a state corresponding to the actual values of OUT and ADDOUT. Two alternatives exist, which differ from each others in case there is a deviation between WSP and MV at the moment of switching over. The step implied by the P part is let through, if BUMPLESS is chosen to 0, but not if BUMPLESS is chosen to 1. Therefore, a transition with BUMPLESS=1 is more bumpless, even if the setpoint rate limit allows a quick change. (See also Section 4.1.4 Setpoint Generation, Ramping, and Tracking and Section 4.1.6 Mode transitions).

The control algorithm is applied with a sampling period that is a multiple of the cycle time of the PC element (the control module). Possible multiples are 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, and 2048. The autotuning automatically selects a suitable sampling period, based on the characteristics of the process. The adaptive control does so too, but restricted to three sampling rates: the saved one last restored (normally obtained by autotuning) and the two closest to that one. During the autotuning the selection of the three rates is determined by the relay phase, and they remain the same until next time a setting is restored.

4.1.3 Scaling and Units

There are ranges defined for the measured value MV, the output OUT, and the scheduling input SCHEDIN. These definitions are made in the data base element, by max and min properties for each of the variables. The ranges define plot ranges in operator displays. They also define the conversion between physical units and percents. Internal calculations in the control algorithm and in the adaptation use values expressed in percents, but PC element inputs and outputs as well as limits are expressed in physical units. The exception is DEV and its limits. They are expressed in percents. The situation is explained by Figure 4-2.

The range defined for the measured value also applies to the setpoint and its limits. The data base properties MAX and MIN define this range. The conversion parameters used in the PC element are redefined, if MAX or MIN is changed in the data base element, using the configuration tool. But not if they are changed by PC connections made to these properties. Changes by PC connections will only affect the graphic presentation of values at the operator station.

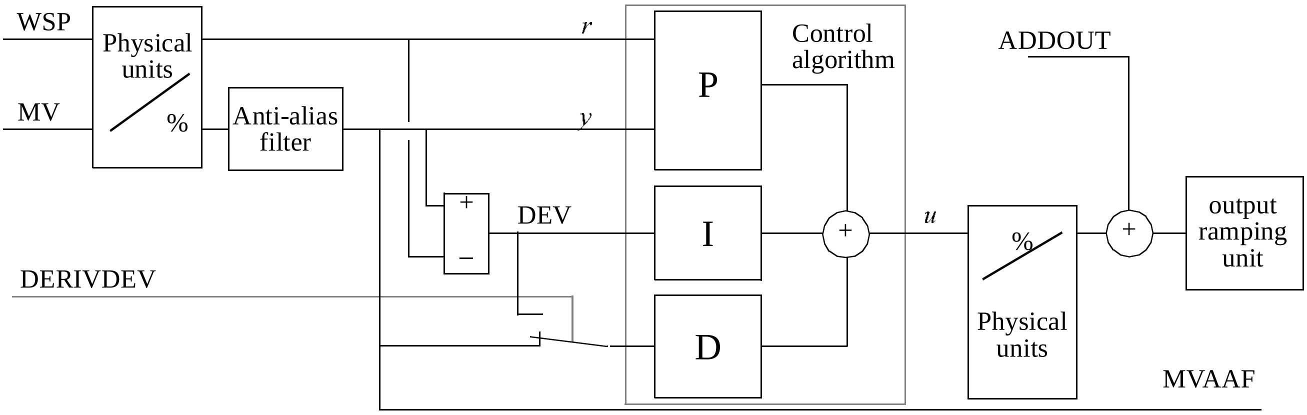


Figure 4-2. The conversion between physical units and percents.

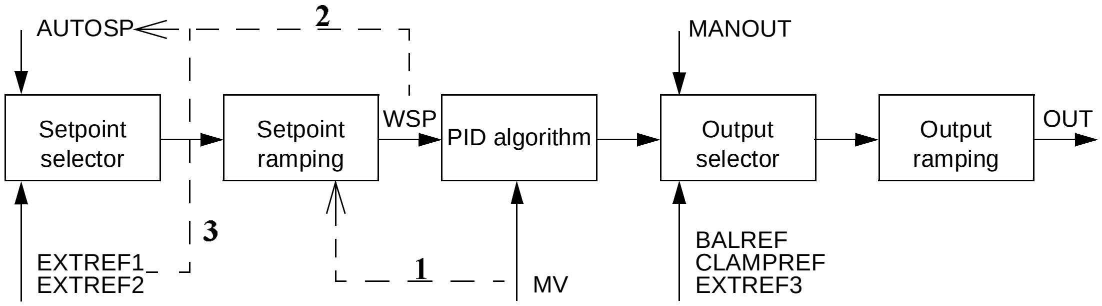
The data base properties OUTMAX and OUTMIN define the range for the output OUT. There are the same conditions for changes as those for MAX and MIN, described above. SCHMAX and SCHMIN define the range for SCHEDIN, only used for the graphic presentation and in the determination of the hystereses used at zone limits.

"Overrange" is allowed in the sense that values may occur outside the 0…100% ranges, but such values will be outside the plot ranges in the operator station. The bottom line is that the max and min properties do not limit the signal values. They merely define the scaling.

If the physical range of MV or OUT is changed using the configuration tool, for a control loop that has already been tuned, the GAIN of the control algorithm will have to be changed, for example by making a new autotuning. But if you are only changing measuring units and thereby the values of MAX and MIN or OUTMAX and OUTMIN, while the physical meaning of the range remains the same, then there is no reason to change the value of GAIN. As an example, suppose you have a measurement range 140…320°F, which means that you have MIN=140 and MAX=320. If you decide to use Celsius instead of Fahrenheit, you will change to MIN=60 and MAX=160. Since 60…160°C is physically the same as 140…320°F, the change will not require any change of GAIN, since 0…100% means the same in both cases, physically. The change of MAX and MIN in PIDCONA will have to be coordinated with a change of MAX and MIN of the corresponding analog input.

4.1.4 Setpoint Generation, Ramping, and Tracking

As determined by the mode, three different sources of the setpoint value are available. One of them is the operator supplied setpoint, AUTOSP. This sections describes how the working setpoint, WSP, is formed and how the operator setpoint may track another setpoint value or the measured value, when not used. Before getting into details, an overview of setpoint tracking possibilities is given in Figure 4-3.



1 WSP may track MV in mode BAL, MAN and E3
2 AUTOSP may track WSP in all modes except AUTO
3 AUTOSP may track EXTREF1 or EXTREF2 in mode E1 or E2 respectively

Figure 4-3. Setpoint tracking options

The setpoint value shown in the operator displays during automatic modes is the working setpoint, WSP. In case AUTOSP is different, it is shown with a minor indication in bar graphs. During non-automatic modes, the operator displays show AUTOSP, since this is the target value that the controller will use if the mode is changed to AUTO. The values EXTREF1 and EXTREF2 are shown in the object display, making it possible to check them prior to a switch-over to mode E1 or E2.

The working setpoint WSP is obtained via a selector and a ramping unit, as shown in detail in Figure 4-4. This figure describes the function in normal operation, not in autotuning (which is described in Section 4.1.7 Autotuning). In all modes, the working setpoint is limited by the limits SETPL and SETPH, given from the operator station. They also limit the value the operator can enter as setpoint, AUTOSP. The actual rate limit depends on the mode as shown in the figure. It limits the speed of a change of the working setpoint, whether the change depends on a change of mode or a change of the value of the reference input. In non-automatic modes WSP is not used in the control algorithm, but it still gets a value, and this without rate limit.

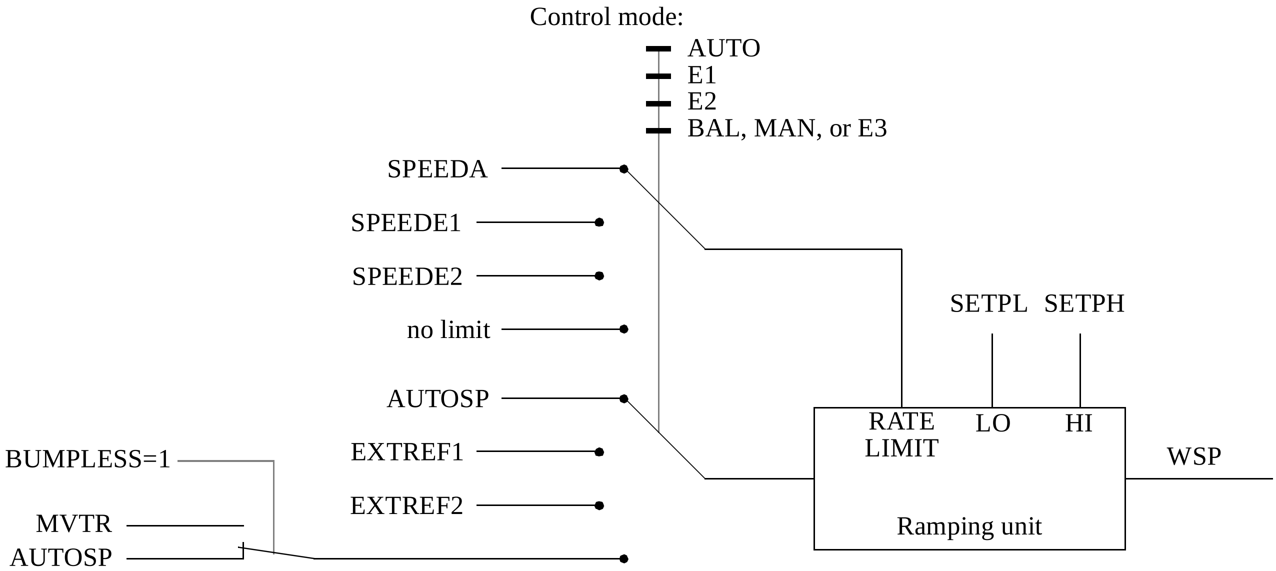


Figure 4-4. Setpoint generation and ramping in normal operation

If the PC element input BUMPLESS is 1, WSP will track the value MVTR in non-automatic modes. It provides a start value for ramping towards the new setpoint, when switching to an automatic mode. MVTR is the anti-alias filtered measured value, with a correction added in case the controller setting does not include integral action (i. e. a P or PD setting). This gives the smoothest possible change-over to automatic modes.

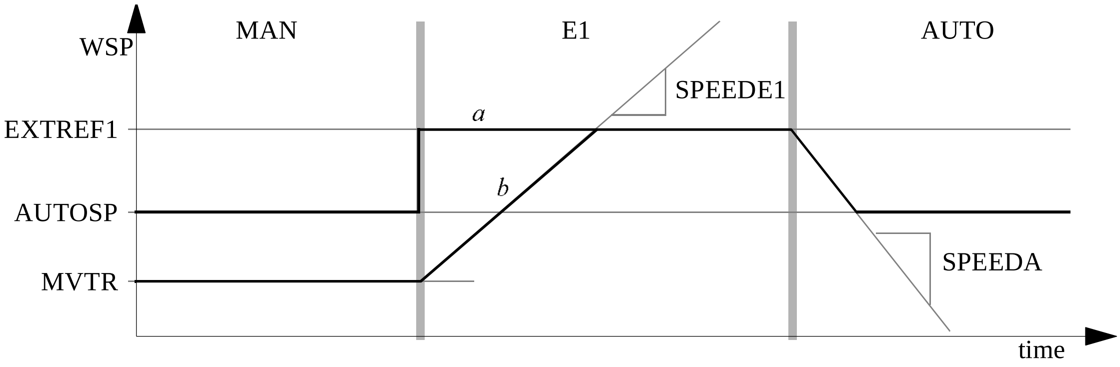


Figure 4-5. An example showing the working setpoint ramping with BUMPLESS = 0 (a), and BUMPLESS = 1 (b), when changing mode from MAN to E1 and then to AUTO.

If BUMPLESS is 0 when switching over from a non-automatic to an automatic mode, the new rate limit is not applied in the very first sample. Thus the working setpoint directly gets the new reference value, AUTOSP, EXTREF1, or EXTREF2. Further changes of the setpoint are of course subject to the rate limits, and this applies to changes between the different automatic modes, AUTO, E1, and E2 as well. The effect is as shown in Figure 4-5.

The operator supplied setpoint is denoted by AUTOSP. This is the target setpoint while in mode AUTO. It is possible to let it track other values while in other modes. This is selected with the PC element inputs TRACKMV and TRACKEXT. They were both assumed to be 0 in Figure 4-5. Otherwise AUTOSP would have tracked either MVTR while in mode MAN, or EXTREF1 or WSP while in mode E1. The tracking options for AUTOSP are shown in Table 4-5.

Table 4-5. The tracking applied to the operator's setpoint, AUTOSP. The shading indicates the tracking reference used, if any. (— means that the value has no influence in that mode)

Conditions			**Tracking reference**			
Mode	**TRACKMV**	**TRACKEXT**	**MVTR**	**WSP**	**EXTREF1**	**EXTREF2**
BAL, MAN, E3	0	—				
BAL, MAN, E3	1	—	▨			
E1, E2	—	0				
E1, E2	—	1		▨		
E1	—	2			▨	
E2	—	2				▨
AUTO	—	—				

While AUTOSP is not tracking, it is available for changes by the operator. This is always the case in mode AUTO. The choice of a non-zero TRACKMV or TRACKEXT, allows a change-free transition from other modes to AUTO. If they are zero, the setpoint value to use (AUTOSP) can be given by the operator (and should be given, or at least checked by a glance at the SP bar graph) before switching the mode to AUTO. With TRACKEXT = 2, the target setpoint value obtained in mode E1 or E2, will still remain the target, if a switch-over to mode AUTO occurs while ramping towards it (but the continued ramp will have a different slope if SPEEDA differs from the rate applied before, which was SPEEDE1 or SPEEDE2). It is only when the switchover to AUTO occurs while the working setpoint, WSP, is ramping towards the external reference, EXTREF1 or EXTREF2, that there is any difference between the choices TRACKEXT=1 and TRACKEXT=2.

4.1.5 Output Generation and Ramping

The selection of value for the output of the controller is shown in Figure 4-6 (the PC element output "OUT"). There is a ramping unit restricting the value to the allowed range and the limited rate of change. The high and low limits are defined either from the operator station, or by PC element inputs (in the later case called "external limits"). The rate limit is defined by the PC element input SPEEDO. As seen in the figure, the input to the ramping unit is always checked against the rate limit, except in mode BAL. After that it is checked against the low limit, and finally against the high limit. This means that in case of conflict, the high and low limits win over the rate limit, and the high limit wins over the low limit.

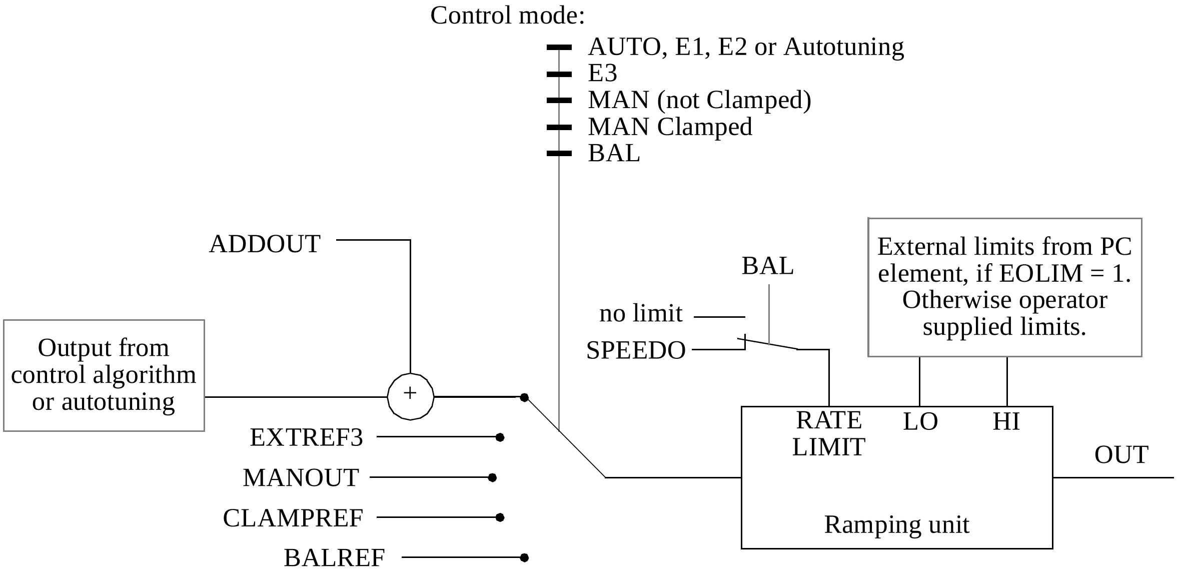

Control mode:
AUTO, E1, E2 or Autotuning
E3
MAN (not Clamped)
MAN Clamped
BAL
ADDOUT
Output from control algorithm or autotuning
+
EXTREF3
MANOUT
CLAMPREF
BALREF
BAL
no limit
SPEEDO
External limits from PC element, if EOLIM = 1. Otherwise operator supplied limits.
RATE LIMIT
LO
HI
Ramping unit
OUT

Figure 4-6. Output selection and ramping

The input to the ramping unit depends on the mode, and whether it is normal operation, or autotuning is going on, as illustrated in Figure 4-6. In automatic modes, the output of the control algorithm is used (after conversion to physical units), with ADDOUT added. The addition of ADDOUT is updated in every execution of the PC element, thereby taking faster variations of ADDOUT into account. The ramping is also executed with the PC element's cycle time. This may seem obvious but it is pointed out since, in contrast to ADDOUT and the ramping, the control algorithm may generate values at a slower rate. Its sampling period (TS) is an integer multiple of the PC element's cycle time.

During autotuning (independent of the underlying control mode), the output of the autotuning (formed differently in the different phases), replaces that of the control algorithm. The value of ADDOUT is added, possibly varying between the sampling instances used by the autotuning, as in normal operation in automatic modes, and the ramping is also applied as in normal operation.

When in mode MAN (except "MAN Clamped"), the operator gives the target value MANOUT for the output OUT. In all other modes (including "MAN Clamped"), MANOUT tracks the present value of OUT, as illustrated in Figure 4-7. This ensures bumpless transfers from other modes to MAN. The output stays unchanged until you give a new value to MANOUT. You can not enter a value of MANOUT outside the high and low limits valid for OUT

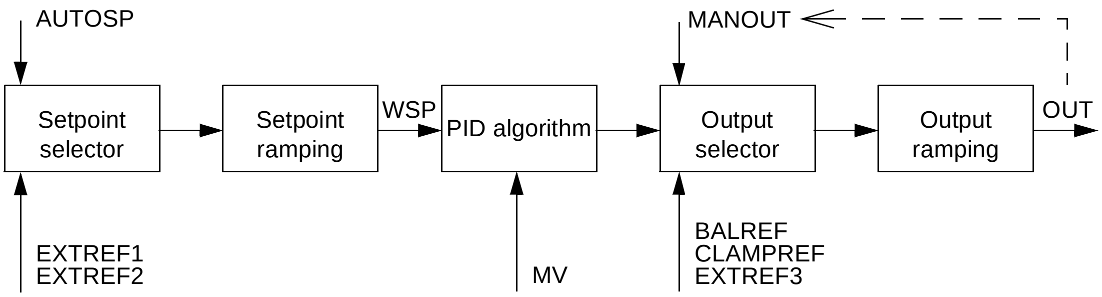


Figure 4-7. The tracking of MANOUT (dashed line) ensures bumpless transitions from any mode to manual mode

4.1.6 Mode transitions

The mode transitions are bounce-free as described in Section 4.1.4 Setpoint Generation, Ramping, and Tracking, Section 4.1.5 Output Generation and Ramping, and Section 4.1.2 Control Algorithm. The exact behavior is affected by the PC element inputs BUMPLESS, TRACKMV, TRACKEXT, SPEEDA, SPEEDE1, and SPEEDE2. Figure 4-8 through Figure 4-10 show examples of transitions to mode AUTO, to illustrate the effect of different choices of the parameters BUMPLESS and SPEEDA. The parameter TRACKMV is zero in these examples, to let the setpoint, AUTOSP, be set to a value different from the measured value, MV, before the transition.

Transition from MAN to AUTO at the time indicated by the arrow

BUMPLESS = 0

Setpoint = 70

The value of SPEEDA has no influence over this transition

A “P step” is obtained, just as with a setpoint step in mode AUTO

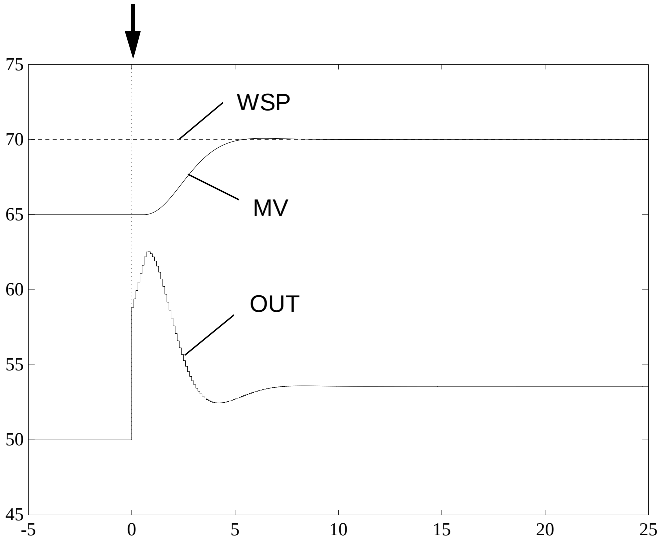


Figure 4-8. Conditioned transfer

With BUMPLESS=0, the control algorithm will be initialized to resemble the conditions during steady state with feedback.The conditioned transfer obtained then gives the same kind of transient as if a setpoint step was made in automatic mode (Figure 4-8).

Transition from MAN to AUTO at the time indicated by the arrow

BUMPLESS = 1

Setpoint = 70

The value of SPEEDA allows a setpoint step of 5 in one sample

There is no "P step"

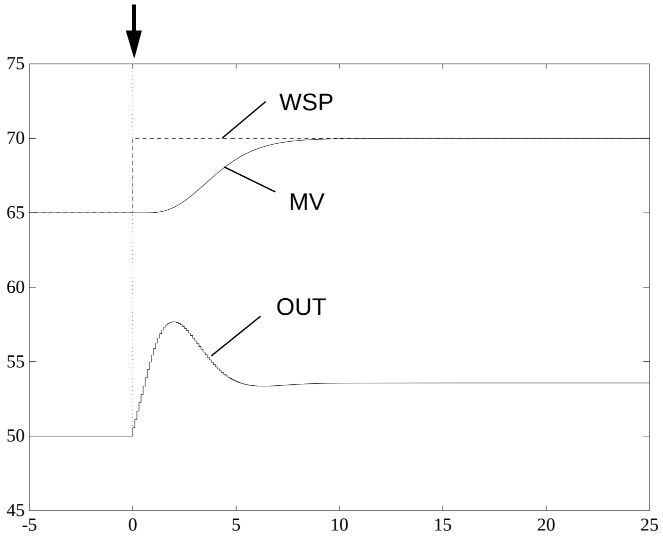


Figure 4-9. Bumpless transfer

With BUMPLESS=1 you get a bumpless transfer (Figure 4-9) which can be made even smoother by restricting the setpoint rate of change (Figure 4-10).

Transition from MAN to AUTO at the time indicated by the arrow

BUMPLESS = 1

Setpoint = 70

The value of SPEEDA limits the setpoint rate of change to produce a ramp

There is no "P step"

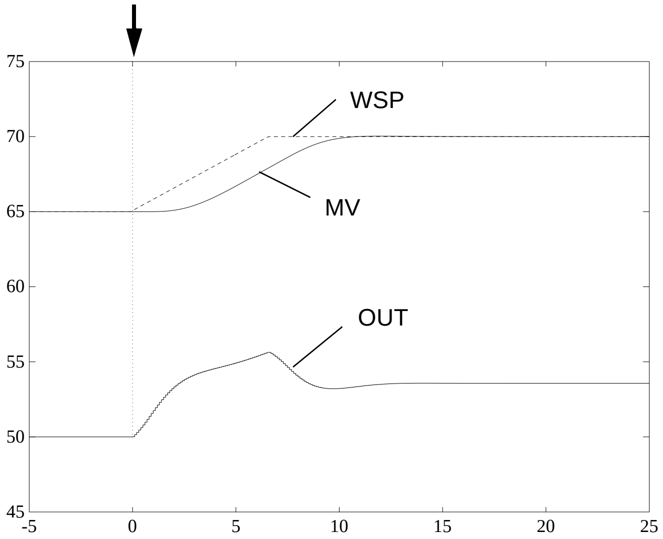


Figure 4-10. Bumpless transfer with a ramped setpoint

In the operator displays, the chosen setpoint value is shown in bar graphs and numerically, while still in manual mode. After the transition to automatic mode, the working setpoint becomes more significant, and this is then the value shown in the bar graphs as well as numerically. During periods in automatic modes when the operator setpoint AUTOSP differs from the working setpoint WSP, the former is shown with a minor indication in bar graphs. It will, for example, indicate the ramp target value during the WSP ramp in Figure 4-10.

It is always possible to check the setpoint value to be applied before a switch-over to an automatic mode. Beside the operator setpoint treated above, the external reference setpoints (EXTREF1 and EXTREF2) are available in the object display.

4.1.7 Autotuning

Autotuning is a method to get the controller tuned automatically. In contrast to adaptive control, autotuning is a **one-shot** tuning, initiated by the operator. The results of the autotuning are a controller setting which will be used in the control algorithm and a process model which allows the continued control to be adaptive.

The autotuning brings the control loop through a sequence of four distinct phases, the Listen phase, the Relay phase, the Excited adaptive control phase, and the Verification phase, in this order. To show the progress, the phase is indicated in the operator's object display.

During the **Listen phase**, the output remains constant, and some values needed for the rest of the autotuning are determined, based on the observed measured values. Among them its mean value. If the selected control mode is MAN or E3, this mean value will be used as setpoint in the later phases of the autotuning.

The Listen phase lasts as long as needed for getting an initial estimate of the size of the variations in the measured value. With a configured cycle time of one second, this typically takes between a quarter of a minute and one minute. During this time the controller output is kept constant, and the spontaneous process variations are monitored. Based on these variations, it is determined which sampling rate, which filter and which relay hysteresis to use initially in the next phase. It may take longer, if there are big variations. But there is also supervision, so that a big deviation will force the autotuning to leave the Listen phase. This happens if the filtered deviation exceeds limits that depend on the initial deviation and the specified setpoint excitation level.

During the **Relay phase**, the output is determined by relay feedback of the measured value compared to the setpoint. First an output step is made from the initial level (the one used in the previous phase). When the process has responded (beyond the hysteresis), the output switches over, across the initial level. When the response crosses the setpoint again, beyond the hysteresis, the output switches back, and this way it continues. A periodic oscillation is obtained, as shown in Figure 4-11. The oscillation indicates the time scale of the process. Identification of a model is started, and a controller setting to use in the next phase is found. The same setpoint value is used throughout the whole Relay phase. It is determined at the moment of leaving the Listen phase. In modes MAN and E3 it will be the mean of the measured value. In modes AUTO, E1, and E2 it will be the normal value for the setpoint in that mode (except that rate limits are ignored during the Listen phase).

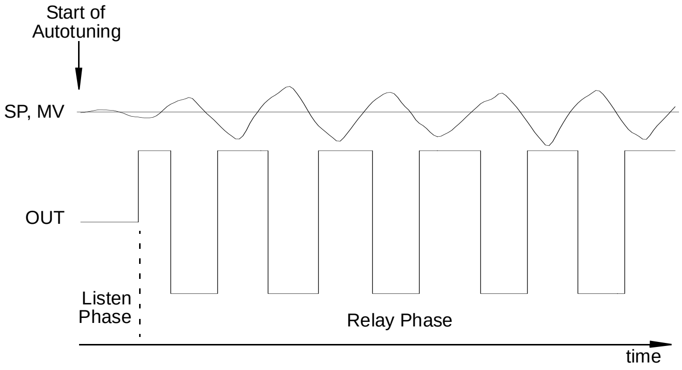


Figure 4-11. The first two phases of the autotuning

The process is controlled during the Relay phase, with the relay function acting as a primitive controller. The number of output steps required for the phase is often around 10 or less, but more may be required, especially in a Controller with high load, or if it is hard to model the process for any reason (e. g. a high disturbance level, non-linearities, or varying dynamics). The time between the steps is entirely determined by the response from the process. It is the feedback via the relay that causes self-oscillation with a frequency that is determined by the dynamics of the process (its time scale).

During the **Excited adaptive control phase** (or shortly the **Excitation phase**), the control algorithm is used with the present setting, updated by adaptation. The setpoint used in the algorithm is a basic setpoint value with excitation added. The excitation is in the form of a PRBS (Pseudo Random Binary Sequence), which means that there are step pulses of varying length. The excitation is not subject to the rate limits normally applying to setpoint changes. The basic setpoint value is obtained the normal way if the mode is AUTO, E1, or E2. It may vary. If the mode is MAN or E3, the basic setpoint value stays constant. If the mode is changed, from AUTO, E1, or E2, to MAN or E3, the basic setpoint remains unchanged. The phase continues until the controller parameters have converged. Tighter limits for the controller output are used initially in this phase. The limits are at first equal to the levels used in the Relay phase. As soon as the setting fulfills certain robustness checks, the limits are gradually opened to the limits normally used. These gradually opening limits are indicated by dashed lines in Figure 4-12, an example where the robustness criteria were fulfilled directly in the Excitation Phase.

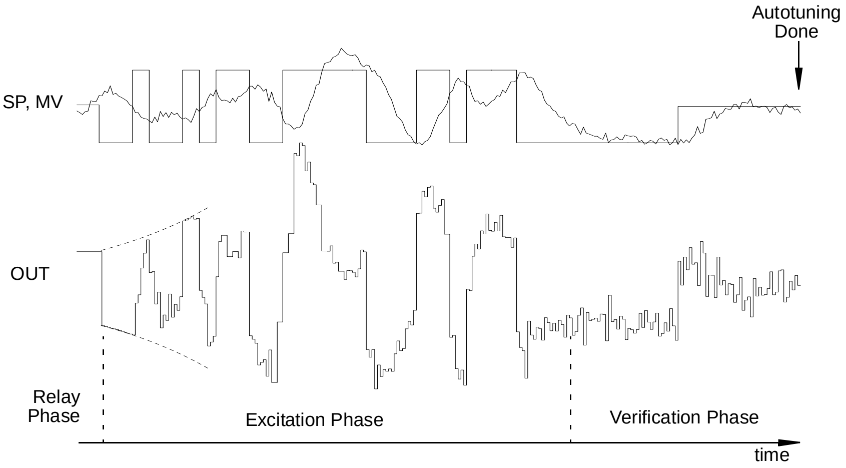


Figure 4-12. The last two phases of the autotuning (Excite and Verify)

NOTE

The process is under PID control throughout the Excitation phase. The differences from normal adaptive control is the tighter initial output limits, until robustness has been assured, the setpoint steps (excitation) added to improve model identification, and the fact that another phase will take over when the setting has converged. If the number of setpoint steps reaches a time-out of 50 without having a converged setting, the autotuning will still switch phase and try to verify the last setting.

During the **Verification phase**, the setting obtained from the Excitation phase is used in the control algorithm. After settling of the measured value, the setpoint excitation is removed in a step. No other changes of the setpoint are let through. The settling and the step response are monitored for agreement with those expected. In this phase, the control of the process is equivalent to what it will be after the autotuning if the present setting is accepted. After the Verification phase, there are three alternatives:

- The autotuning is finished. Either the verification was successful, or the decision left to the user was responded with an "Accept and Save" command.
- The Excitation phase is resumed. Either the verification failed, or the decision left to the user was responded with a "Continue Tune" command.
- The autotuning is aborted. Either the verification failed after a very long period of Excitation phase and Verification phase execution (step number time-out), or the decision left to the user was responded with a "Stop and Restore" command.

As indicated above there are cases when a user decision is required. A high noise level, compared to the excitation, may for example prevent the verification from distinguishing the responses from the effect of the noise, and then the decision is up to the user instead. A user decision is also required, if gain scheduling is enabled and the responses accepted by the verification were obtained operating in another zone, than the one where the autotuning was started. While waiting for the user decision, the control goes on with the same present parameter setting and a constant setpoint.

The number of setpoint steps are counted in the excitation phase and the verification phase together. It is only when this number has reached time-out (50), that the autotuning aborts itself in case the verification fails. Otherwise the excitation phase is resumed if the verification fails.

When the autotuning is finished, the setting and its corresponding model are saved. If gain scheduling was enabled when starting, the autotuning is considered to belong to the zone where it was started. Each zone has its own place for its saved setting. The autotuned setting is thus saved at the place for the start zone. While the autotuning session is still going on, it is possible to check in the object display, which zones have been visited during the session. Without gain scheduling enabled, the place of zone 1 is used for the saving.

A complete autotuning is started by selecting "Start Tune" in the maintenance level dialog “Autotune…”. This means that the autotuning will start from the beginning, i. e. with the Listen phase.

A shorter autotuning can also be performed, starting directly in the Excitation phase. This requires that the saved controller setting for the start zone was obtained by autotuning or adaptive control, i. e. it has an associated model. This shorter autotuning is started by selecting “Continue Tune” in the maintenance level dialog "Autotune…".

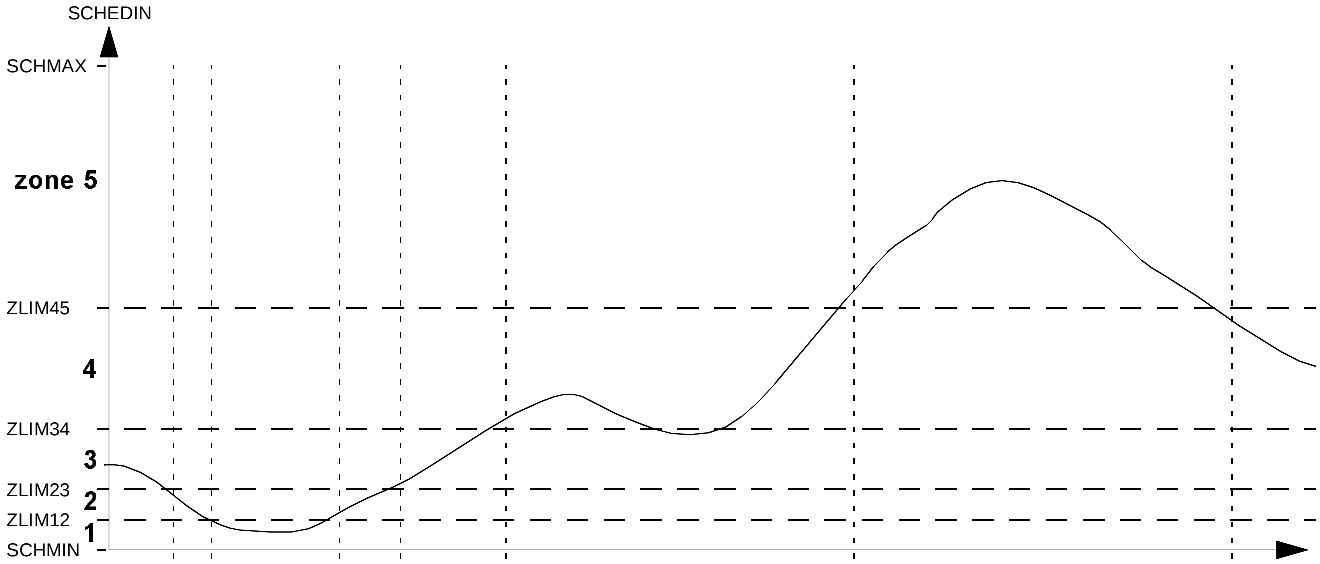
4.1.8 Gain Scheduling

PIDCONA is provided with gain scheduling, or rather parameter scheduling. Assuming the extension C6 of the PC element is included, there are five sets of parameters (controller settings) corresponding to five zones of the scheduling variable, the PC element input SCHEDIN. The gain scheduling is enabled or disabled with the input GSENBL. Operation without gain scheduling enabled is treated as being in zone 1. For a configuration example, see Section 3.4.2, Configuration example with Gain Scheduling.

When gain scheduling is enabled, and SCHEDIN travels across the zones, the setting is switched between the parameter sets. Thereby a setting tuned for the current zone is always used. In fact, it is accomplished the following way. When passing a zone limit with the gain scheduling enabled, the parameters of the new zone are switched in, to become the present setting used in the control algorithm (also shown as present parameters in the object display). The number of the current zone is also available as a PC element output, CURRZONE. In case there was no setting saved for the new zone entered, there is no change of the present setting. Further information is given in Section 4.1.10 Present and Saved Controller Parameters, on saved settings and object display indications for them.

The zones are defined by the four limits, the PC element inputs ZLIM12, ZLIM23, ZLIM34, and ZLIM45. There is a hysteresis around each zone limit, to avoid unnecessary switching. It is automatically adjusted in proportion to the width of the zones, but it is never smaller than 1% of the range defined for SCHEDIN.

To illustrate how the gain scheduling works, Figure 4-13 shows the development over time of the scheduling input, SCHEDIN. This is essentially how it would look for the example of Section 3.4.2, Configuration example with Gain Scheduling when all zones have been autotuned. You can see how the current zone number and present settings change when SCHEDIN crosses the zone limits. In the middle of the time shown, while in zone 4, the limit ZLIM34 is crossed for a while, but not beyond the zone limit hysteresis, hence the zone is not changed.



Zone	3	2	1	2	3	4	5	4
Gain	11	2.3	0.65	2.3	11	40	245	40
TI	76	114	150	114	76	55	31	55
TD	18	20	23	20	18	15	10	15
Beta	0.39	0.43	0.47	0.43	0.39	0.39	0.38	0.39
TS	4	8	8	8	4	4	2	4
Frequency	0.05	0.03	0.02	0.03	0.05	0.07	0.12	0.07

Figure 4-13. The variations of the controller parameters with gain scheduling

A change of GSENBL from 0 to 1 may lead to a change of the current zone (CURRZONE) from 1 to something else, but even so, it does not change the present setting. In applications where that is desired, the input to GSENBL would be connected to the input RESTORE as well.

The schedule is most easily built by autotuning the loop in each of the zones. The setting obtained in an autotuning session belongs to the zone where it was started. The mechanism of the gain scheduling (to switch to a new setting when a zone limit is passed) is of course not active during autotuning, and the zone number indicated in the object display as well as the PC element output CURRZONE show the start zone throughout the whole autotuning session. If the time scale of the process does not differ too much between the zones, a setting obtained previously for one zone can be used as a starting point for the other zones, using the shorter autotuning started by "Continue Tune". This requires that the present setting is retained, while SCHEDIN passes from the zone having the desired start setting to the zone to be tuned. This is the case if either there is no previous setting stored in that zone or the gain scheduling is disabled when SCHEDIN enters the new zone. In the latter case gain scheduling must be re-enabled before starting the tuning, unless zone 1 is the one to be tuned.

4.1.9 Adaptive Control

Adaptive control means that the controller setting is adjusted automatically to changes of the process dynamics, and that these adjustments are not preprogrammed as in the case of gain scheduling. Instead some kind of identification of the dynamics is needed. With adaptive control in PIDCONA, a model of the process is kept up-to-date by recursive identification, and the controller setting is obtained by a design method called dominant pole placement, applied to this model. It takes into account the response type specified by the user, and the frequency limit, if given. That is just as in autotuning. The difference from the Excitation phase of the autotuning is that no extra excitation is added, and there is no limit in time, how long the adaptive control can go on.

Adaptive control can very well be used together with gain scheduling. On entering a new gain scheduling zone, adaptation will start over from the setting saved for that zone.

A requirement for using adaptive control is that autotuning has been done, so that there is a model available to start from. With gain scheduling, the control will be adaptive only in the zones where a model is associated with the saved setting, not in zones where it was tuned manually.

There is no adaptation in non-automatic control modes.

To summarize, the control is adaptive if

- adaptation is enabled in the operator station (maintenance level dialog "Adaptation…", key "Adaptation Enable"), and
- the control mode is automatic (AUTO, E1, or E2), and
- the PC element input ADAPON is 1, and
- both the present setting and the saved one (in the current zone) were autotuned or adapted so that they have an associated model (thus marked (A) in the object display).

There are supervisions built in, which restore (revert to) the saved controller setting, in case trouble with the adaptation is detected. It is also possible to make application-dependent decisions to perform a restore, using AMPL to provide a positive transition at the PC element input RESTORE.

4.1.10 Present and Saved Controller Parameters

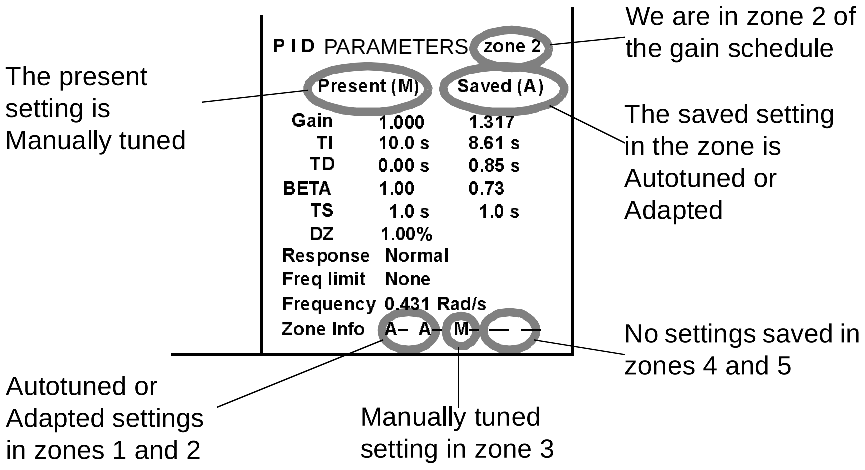
In the object display, two sets of controller parameters are shown, the present values and the saved values. As mentioned in Section 4.1.8 Gain Scheduling, there are five sets of saved parameters, one for each gain scheduling zone. The set shown is the one for the current zone.

When tuning the controller manually, values for GAIN, TI, TD, BETA, and TS are entered in the operator's dialog "Parameters…" in maintenance level. They directly change the present parameter set. When autotuning, the result is placed both in the present parameter set and in the saved parameter set for the zone concerned (cf. Section 4.1.7 Autotuning). During adaptive control, the present parameter set is affected by the adaptation, while the saved parameters remain unchanged, available as fall-back settings. With gain scheduling, the present parameters are changed by copying saved parameters when zone limits are crossed. However, if there is no setting saved in the newly entered zone, the present parameters remain unchanged.

There are also the following operator commands in the maintenance level dialog “Adaptation…”: "Save" copies the present set to the saved set of the current zone, and "Restore" copies the saved set of the current zone to the present set. A positive transition (0→1) of the PC element input RESTORE has the same effect as the operator command "Restore". If tuning is done manually, the resulting parameter setting is saved using the "Save" command.

Autotuned or adapted settings contain some more parameters in addition to those mentioned above. There is a frequency value, holding information about how fast the closed loop is made, there is the response type used (in "Response Setup…" dialog), and there are model parameters. Of these, the response type and the frequency value are shown in the object display at the operator station. The frequency value is meant to give a hint about the relevant magnitude, when specifying a frequency limit ("Extended Setup…" dialog). The presence of these extra parameters is indicated by an A (for Autotuned or Adapted), while the absence is indicated by an M (for Manual tuning).

If you have gain scheduling (PC element extension C6), the line "Zone Info" in the object display indicates what kind of settings are saved, with one character for each of the five zones. A dash (–) indicates that no setting was saved. An example is shown in Figure 4-14. (A second character for each zone indicates, during an autotuning, whether the zone was visited or not.)



PID PARAMETERS zone 2

	Present (M)	Saved (A)
Gain	1.000	1.317
TI	10.0 s	8.61 s
TD	0.00 s	0.85 s
BETA	1.00	0.73
TS	1.0 s	1.0 s
DZ	1.00%	
Response	Normal	
Freq limit	None	
Frequency	0.431 Rad/s	
Zone Info	A– A– M– — —	

The present setting is Manually tuned
We are in zone 2 of the gain schedule
The saved setting in the zone is Autotuned or Adapted
No settings saved in zones 4 and 5
Autotuned or Adapted settings in zones 1 and 2
Manually tuned setting in zone 3

Figure 4-14. Object display information on present and saved parameter settings

4.1.11 Alarms, Limitations and Events

The subject of limits for the setpoint and the output has been treated in Section 4.1.4 Setpoint Generation, Ramping, and Tracking and Section 4.1.5 Output Generation and Ramping.

The measured value and the deviation are supervised against alarm levels. These levels are defined at the operator station, MV L2, MV L1, MV H1, MV H2, DEV L1, and DEV H1. Around each of these levels there is a hysteresis for the alarm, MVHYST or DEVHYST. These hystereses are defined in the data base element.

The measured value is anti-alias filtered before the supervision. The anti-alias filtered measured value is called MVAAF. It is available as a PC element output as well, and this is the value shown as measured value in the operator displays. The anti-alias filter is adjusted to the present sampling period, TS, which may vary from zone to zone with gain scheduling. It may also vary in a restricted range during adaptive control.

Model based supervision of the deviation is used, while the present controller setting has an associated model (i. e. if it was obtained by autotuning or by adaptive control). This means that the deviation from the expected response to setpoint changes is used. Otherwise the deviation DEV is used. The value DEV (PC element output and value presented in operator station) is always calculated as shown in Figure 4-2, also when the supervision is model based.

The events handled by PIDCONA are divided into the following five types, with different event treatment:

1. The measured value passes low level MV L1 or high level MV H1, or the deviation passes low level DEV L1 or high level DEV H1.
2. The measured value passes low level MV L2 or high level MV H2.
3. Control mode is changed by one of the "flash mode" PC element inputs (LOCALFL, MANFL, AUTOFL, E1FL, E2FL, or E3FL).
4. Error detected at the analog input channel, AI error alarm.
5. Events related to autotuning or adaptation.

Suppose the default values were used for the event treat pointers in the data base element, H1L1_TR, H2L2_TR, MODE_TR, ERR_TR, and ADAP_TR, respectively. Then an alarm requiring acknowledgment is generated when an "error" event of type 1, 2, 4, or 5 occurs. It is also registered in both the alarm list and the event list. An event of type 3 is only registered in the event list. A "return to normal" event of any type 1, 2, 3, 4, or 5 is also registered, but only in the event list.

It is possible to block the treatment of the events of type 1 or 2 by operator dialog, or by PC connections to the data base properties ABL_H1L1 and ABL_H2L2, or by the PC element input ALCBLK. If blocked by ALCBLK=1, not only the event handling is blocked, but the status outputs of the PC element, MV>H2, MV>H1, MV<L1, MV<L2, DEV>H1, and DEV<L1, are also forced to zero. When the alarms are blocked by dialog, or by data base properties ABL_H1L1=1 or ABL_H2L2=1, the blocking is indicated in the operator display. These ways of blocking the event handling do not affect the PC element outputs.

The printout of events of type 1 and 2 can be blocked separately, by operator dialog, without affecting any other parts of the event handling.

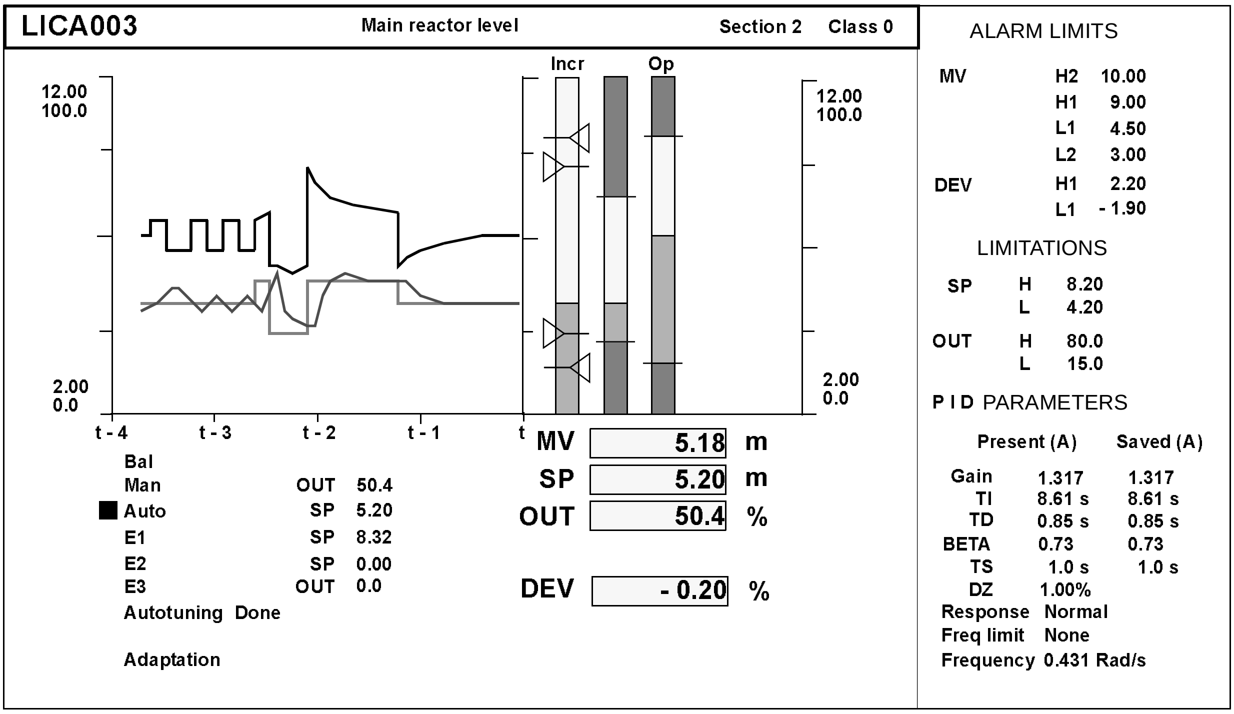
4.2 Operating Overview

The Operator Station provides presentation of the PIDCONA objects and dialogs for the manipulation, and there are functions for the event handling. These are all standard functions in the control system.

In everyday operation you get process information via the displays, both those built for your application, and the object display which provides detailed information without need for configuration. The production level dialog allows you to change control mode, to alter the value of the setpoint or the output (depending on the mode), and to acknowledge alarms.

The alarms and events appear in lists in the Operator Station, and they can also be printed directly as they occur. All alarms have distinct indications in the object display, thus making it very clear what the alarm is about. While the alarm list gives you an overview over all present alarms, the object display also gives you an overview over all alarms that apply to one particular object.

The object display gives a complete view of the status of the object, including all limits and the controller parameters, and also the autotuning progress.



LICA003 Main reactor level Section 2 Class 0

Bal
Man OUT 50.4
■ Auto SP 5.20
E1 SP 8.32
E2 SP 0.00
E3 OUT 0.0
Autotuning Done

Adaptation

MV 5.18 m
SP 5.20 m
OUT 50.4 %

DEV - 0.20 %

ALARM LIMITS

MV	H2	10.00
	H1	9.00
	L1	4.50
	L2	3.00
DEV	H1	2.20
	L1	- 1.90

LIMITATIONS

SP	H	8.20
	L	4.20
OUT	H	80.0
	L	15.0

P I D PARAMETERS

	Present (A)	Saved (A)
Gain	1.317	1.317
TI	8.61 s	8.61 s
TD	0.85 s	0.85 s
BETA	0.73	0.73
TS	1.0 s	1.0 s
DZ	1.00%	
Response	Normal	
Freq limit	None	
Frequency	0.431 Rad/s	

Figure 4-15. An example of an object display for PIDCONA

With the authority key, the maintenance level dialog can be accessed. It allows you to alter limits, to block alarms, to tune the loop, either by autotuning or by manual changes of the controller parameters, to select adaptive or non-adaptive control, and to save or restore the setting.

4.3 Runtime Tutorial

You handle the PIDCONA object exactly as you handle other objects. Most of the things you do are the same as with other objects. One thing, though, is specific to PIDCONA. That is the autotuning. Below we describe a typical session exemplifying how the autotuning is performed during commissioning. It is assumed that you are already familiar with the way to handle objects in general within the system.

Before autotuning

Start by bringing up the object display for the control loop to tune. There the trim curves and the autotuning progress information are presented, which is valuable during the tuning.

Then select the object and enter the "Autotune…" dialog. Authority key is needed.

We assume the default excitation levels to be appropriate and we want a normal step response, otherwise we might have done changes in the setup dialogs ("Response Setup…" and "Extended Setup…").

The control mode is MAN, since the loop hasn't been in operation yet. You may have to vary the output manually until the process is in a state and at a measured value level where you want to tune it.

NOTE

> You may check that the configuration of reverse or direct action is correct. Above the bar graph for the measured value there is a text, "Incr" or "Decr". It indicates whether the final process response to an increase of the output is expected to be an increase or a decrease of the measured value. If the indication does not agree with the actual behavior of the process, the functional parameter REVACT of the PC element has to be corrected before you start the autotuning. Cf. Section 3.1.4 Reverse Action or Direct Action. "Incr" corresponds to reverse action and "Decr" corresponds to direct action.

Start the Autotuning

When the measured value has stabilized fairly well on an acceptable level, you press Start Tune. See Figure 4-16.

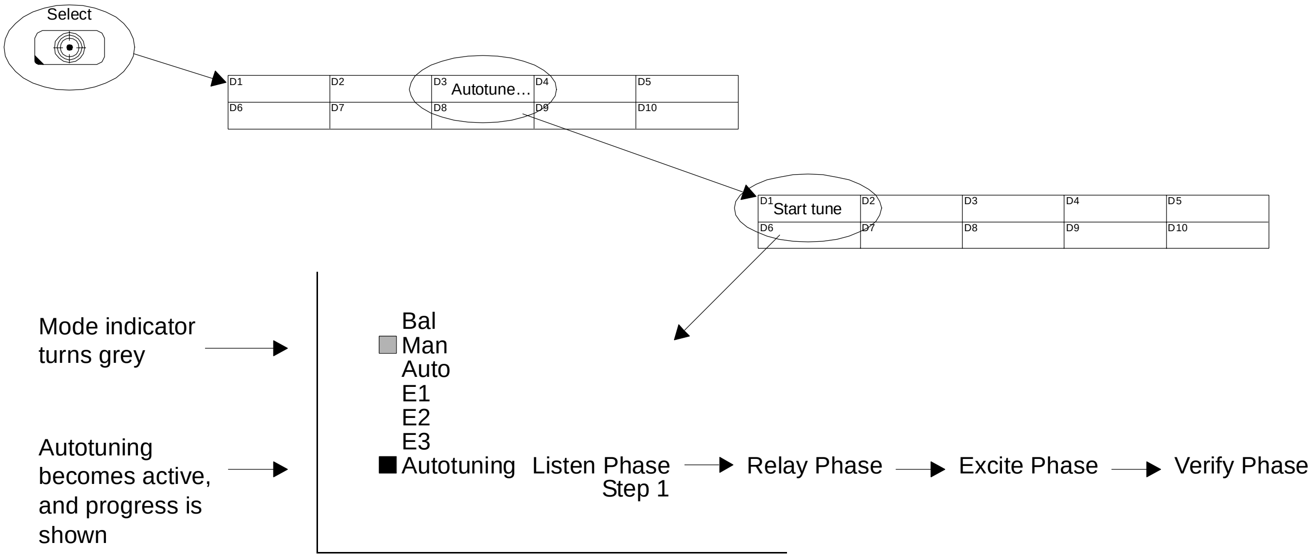


Figure 4-16. Start of automatic tuning

You will see that autotuning becomes active, and the phase is shown, first Listen Phase, and then Relay Phase. The mode indicator turns grey. The actions of the controller are determined by the autotuning, more than by the selected mode. From the grey mode indicator you see what the mode will be after the autotuning. (In the other display elements, autotuning is indicated by "AT" after the mode.)

The Autotuning Progress

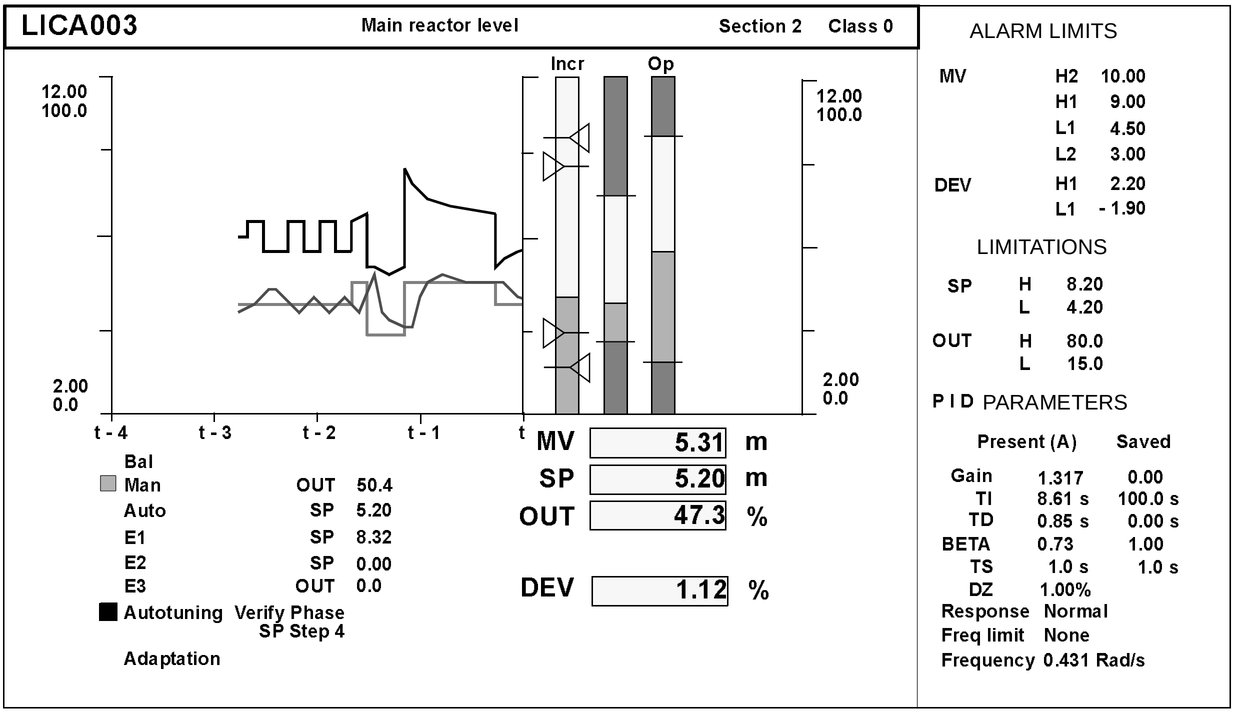
In the relay phase, the output switches between two levels, the initial level plus/minus the output excitation level, the switching instances being determined by the measured value responses. The number of output steps made is shown in the object display.

After a while the autotuning enters the next phase, which is the Excited Adaptive Control Phase (or shortly "Excite Phase"). At the same time new present parameter values are shown. Now the control is adaptive, with excitation added in the form of setpoint steps, in order to help the adaptation to find a final setting. Now the counter value shown in the object display tells you how many setpoint steps are taken.

The parameter value variations are essentially expected to decrease over time, except that they may jump if the sampling period switches to another value. (After the Relay Phase it can switch between three different sampling rates.)

When the parameters have converged, the autotuning enters the Verification Phase. After settling of the measured value at the initial setpoint level, the setpoint returns to its middle value, and the transient is checked and compared to the expected response. This setpoint step also increments the step number shown. The object display may now look like Figure 4-17.

If the verification result was rejected, the "Excite Phase" would be resumed, with further adaptation before a new verification is tried. The autotuning information would show a continued count of the number of setpoint steps.



LICA003 Main reactor level Section 2 Class 0

Bal
▨ Man
Auto
E1
E2
E3
■ Autotuning Verify Phase
SP Step 4
Adaptation

OUT 50.4
SP 5.20
SP 8.32
SP 0.00
OUT 0.0

MV 5.31 m
SP 5.20 m
OUT 47.3 %

DEV 1.12 %

ALARM LIMITS

MV	H2	10.00
	H1	9.00
	L1	4.50
	L2	3.00
DEV	H1	2.20
	L1	- 1.90

LIMITATIONS

SP	H	8.20
	L	4.20
OUT	H	80.0
	L	15.0

P I D PARAMETERS

	Present (A)	Saved
Gain	1.317	0.00
TI	8.61 s	100.0 s
TD	0.85 s	0.00 s
BETA	0.73	1.00
TS	1.0 s	1.0 s
DZ	1.00%	
Response	Normal	
Freq limit	None	
Frequency	0.431 Rad/s	

Figure 4-17. The object display during the Verify phase of autotuning

If the verification result was rejected, the "Excite Phase" would be resumed, with further adaptation before a new verification is tried. The autotuning information would show a continued count of the number of setpoint steps.

Autotuning has Finished

As the step response obtained during the verification had the expected speed and shape, the autotuning is finished. The information line in the object display says "Autotuning Done". The obtained parameter values were copied to the "Saved" set, which is also shown. The control mode is returned to the one that was shown in grey during the autotuning.

Now the loop is ready for operating in automatic control modes.

Extra Information Appears in the Object Display

Consider a case similar to the previous example, except that there are more disturbances. When the "Excite Phase" is entered the following happens:

The disturbance level turns out to be a little too high, in relation to the setpoint steps, to give enough signal to noise ratio for the identification and the verification. Therefore a warning "Low Excitation" appears in the object display. With the range 0 … 12 m, the setpoint varies plus/minus 0.36 m (3% of the range) from a middle value, that was determined when the Listen Phase was completed. Since we ran the Listen Phase in mode MAN, the middle value was taken as the average measured value from the Listen Phase.

When the text "Low Excitation" appears, we decide to increase the setpoint excitation level from the default 0.36 m, see Figure 4-18. When it was increased to 0.5 m the text "Low Excitation" disappeared, so we leave the excitation level at that value.

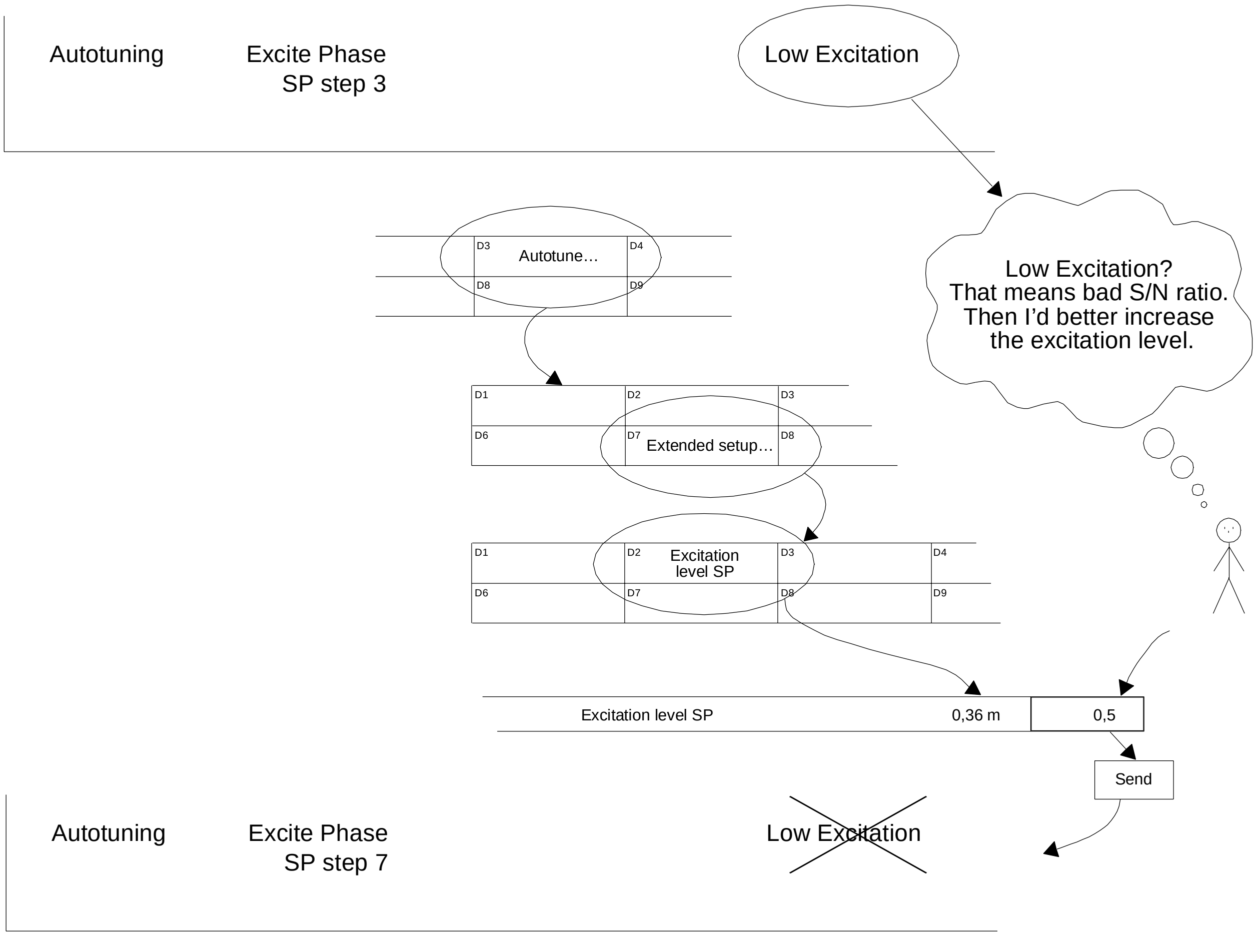


Figure 4-18. Adjustment of the excitation level for the setpoint

The autotuning continues as in the previous example. When the controller parameters have converged in the "Excite Phase", the "Verify Phase" takes over. The autotuning is finished when the measured value has settled in the expected manner, after the setpoint step of this phase.

4.4 Operating Instructions

The general way of viewing, selecting, and manipulating objects is described in *AdvaCommand Basic Functions User's Guide*. There is also a description of common properties for the functional units in *Functional Units Part 1 Summary and Common Properties*.

The description here is concentrated on what you can see and do in the Operator Station, regarding the functional unit PIDCONA. Detailed descriptions of the dialog and the display elements are found in Appendix A, Display Elements and Appendix B, Process Dialog.

4.4.1 Change of Control Mode

The different control modes applicable to PIDCONA are described in Chapter 4.1.1. The present mode is shown in the display elements. In the object display, it is indicated by a green box beside the name of the present mode. In the other display elements an abbreviation of the mode is shown.

Table 4-6. Mode abbreviations used in display elements

Mode	Abbreviation
BAL Local	BaLo
BAL	Bal
MAN Clamped	MCp
MAN Forced	MFd
MAN	M
AUTO	A
E1	E1
E2	E2
E3	E3

The modes available for operator selection are: MAN, AUTO, E1, E2, and E3. There are direct access keys and dynamic keys for all except E3, which only has a dynamic key in the top level dialog. The "mode enable" PC element inputs, one for each of these modes, determine which modes are actually selectable by the operator, for the particular control loop.

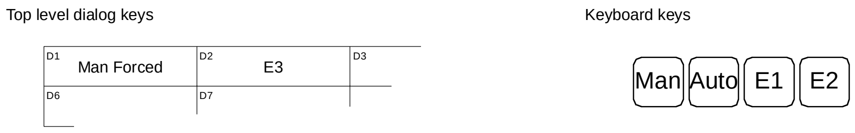


Figure 4-19. Mode keys

You may want to check what setpoint value will be used before the transition to an automatic mode. On leaving a non-automatic mode, the operator setpoint is always available in all display elements that show the setpoint. It is the one that will be valid after a change to mode AUTO. On leaving mode E1 or E2 for AUTO, the operator setpoint is shown with a minor indication in the bar graphs, if it differs from the working setpoint in use. All setpoint values (AUTOSP from the operator; and the external references EXTREF1 and EXTREF2 from other sources) can always be checked in the object display.

Mode changes can also be programmed using PC element inputs, with priority order as described in Table 4-3 and Table 4-4 in Chapter 4.1.1. Some programmed mode commands have higher priority than the ordinary operator keys for mode changes. But in all modes except "BAL" and "MAN Clamped" the operator has a possibility to override the programmed mode and force it to "MAN" with the dynamic key "Man Forced" in the top level dialog. This gives control mode "MAN" with the additional status "Forced". PIDCONA doesn’t leave this status until another mode is selected by the operator. The PC element input MANENBL is required to be set, both for selecting "MAN" and for selecting "Man Forced".

4.4.2 Change of Setpoint or Output

The setpoint value used in mode AUTO is the one that is determined by the operator, AUTOSP. It is denoted by SP in the operator’s dialog. It is always possible to change it, while in mode AUTO. It may also be available for changes while in another mode, thus enabling you to preselect the setpoint value to be used when mode AUTO is entered. This availability depends on the tracking options chosen via PC element inputs TRACKMV and TRACKEXT. While tracking, it is not possible to change the setpoint manually. Instead it is automatically adjusted to a value that makes the mode transfer change-free in the setpoint.

In mode MAN the operator determines the output value of PIDCONA. It is denoted by OUT in the operator’s dialog. The manual changes of OUT can only be done in mode MAN, since in all other modes the output is determined from another source. Also the additional status "Clamped" of mode MAN prevents manual changes of the output. When entering mode MAN (without Clamped status) the output remains at its last value obtained in the previous mode, until it is changed by the operator.

There are two direct access keys for increase and decrease of a value. They operate on either the setpoint or the output, the one indicated as active by the text SP or OUT in the dialog line. The relevant one automatically becomes active on a change to mode AUTO or MAN. They can also be activated the usual way, by their keys SP and OUT, in the top level dialog. When none of them is indicated, the keys still operate on OUT in mode MAN and on SP in mode AUTO.

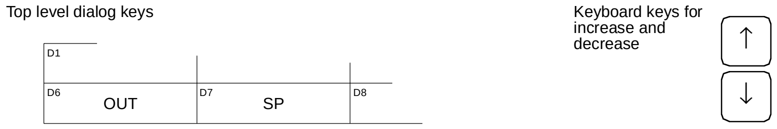


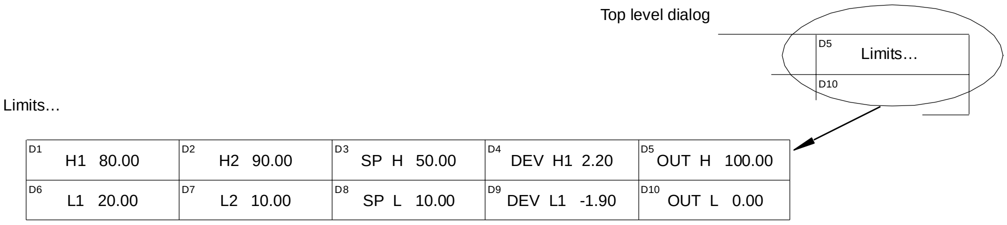
Figure 4-20. Keys for change of setpoint value or output value

With the increase and decrease keys, the value is changed in steps of 0.5% of the range. If the shift key is pressed simultaneously the steps are 5%.

The setpoint and the output can also be altered via the dialog line input field, just like other numerical values. When you make a setpoint change that is bigger than the rate limiter, we give an example. The presentation of the SP value will show what is actually used during the ramping (the working setpoint), while the target value is shown with a minor indication in the bar graph. This minor indication is also used for the operator setpoint if it differs from the working setpoint value, while in mode E1 or E2, or during autotuning.

4.4.3 Limits

Alarm limits and constraint limits for the output and the setpoint in PIDCONA are accessed via the top level dialog choice "Limits…", for changes of the values. The bar graphs for MV, SP, and OUT have graphical indications for the limits, which are highlighted when they are active.



D1 H1 80.00	D2 H2 90.00	D3 SP H 50.00	D4 DEV H1 2.20	D5 OUT H 100.00
D6 L1 20.00	D7 L2 10.00	D8 SP L 10.00	D9 DEV L1 -1.90	D10 OUT L 0.00

Figure 4-21. Dialog keys for change of limits

NOTE

The deviation limits can be chosen tighter when the controller has been automatically tuned, since it then supervises the deviation from the expected measured value using a model of the closed loop behavior. When not automatically tuned, and also during an ongoing autotuning, it is instead the instantaneous deviation of the measured value from the setpoint value that is supervised, that is, the traditional deviation. This is also the one shown in the object display. In this case the supervised deviations become larger during setpoint changes and the limits cannot be chosen as tight, without getting more deviation alarms during setpoint changes.

The rate limit for the output is not available in any operator dialog, but as a PC element input, SPEEDO. However, if there is an actual rate constraint in the actuator, it is important that this PC element input conforms with it. Otherwise, output changes known as steps by PIDCONA, but degraded to ramps by the actuator, will fool the adaptation during adaptive control and autotuning.

4.4.4 Alarms

The alarms generated within PIDCONA are treated in the standard way described in *AdvaCommand Basic Functions User’s Guide*. For a description of how group alarm is handled, you are referred to the same document. Below we give a description of the alarms within PIDCONA, assuming that standard event handling was chosen in the configuration. A complete list of the standard event texts for PIDCONA is given in Appendix E, Standard Event Texts.

With standard event handling, the alarms detected give the following texts in the alarm line, preceded by the name and description of the object, and followed by the date and time. Other texts may be used, if non-standard event treatment pointer values have been selected (in the Event Treat part of the PIDCONA data base element).

Table 4-7. Alarm Line Text

Property Text	Event Text	Condition causing the alarm
MV<L1	<Limit value, unit>	Measured value(1) outside limit
MV>H1	<Limit value, unit>	Measured value(1) outside limit
DEV<L	<Limit value, unit>	Deviation(2) outside limit
DEV>H	<Limit value, unit>	Deviation(2) outside limit
MV<L2	<Limit value, unit>	Measured value(1) outside limit
MV>H2	<Limit value, unit>	Measured value(1) outside limit
AISign Err	Alarm	Error detected concerning the analog input channel
AT Abort	Alarm	Autotuning was aborted
AT Fail	Alarm	Autotuning has failed
AT Quest	Decision?	You have to decide: Shall autotuning proceed?
TS Fault	Alarm	There is an invalid sampling rate
Ad Fail	Alarm	Adaptation has failed during adaptive control

(1) The value supervised is the one after anti-alias filtering, MVAAF, see Figure 4-2 and page 4-20
(2) The deviation from the expected value, or from the actual setpoint, as explained in page 4-20

All alarms also give red indications in the object display, to point out the cause.

With the alarm "AT Quest; Decision?", the question "CONTINUE, ACCEPT or STOP?" appears in the object display, and you have to make your decision: either "Continue Tune", "Accept and Save", or "Stop and Restore" in the "Autotune…" dialog. See further page 4-34.

Advice on actions when autotuning was aborted or failed are given in page 4-34. See also Section 5.4.2, Fault Finding Regarding Autotuning.

The alarms normally require acknowledgment. However, with the three autotuning alarms (AT Abort, AT Fail, and AT Quest), the action normally taken is to restart the autotuning or to make the decision requested. This automatically acknowledges the alarm. You can also acknowledge these alarms with the acknowledge key, before taking action, but it is not necessary.

An invalid sampling rate can occur if you change the cycle time in the Controller, or if you enter a data base that includes previously saved settings. The invalid rate can be in any of the five saved sets of controller parameters. The alarm remains until every set that is saved has a valid sampling period.

An adaptation failure alarm is usually a signal that something in the process is abnormal. See also Section 5.4.3, Fault Finding regarding Adaptive Control.

4.4.5 Alarm Blocking

On maintenance level, the dialog choice "Block…" allows you to block limit alarms. It is done separately for type 1 alarms and for type 2 alarms, and separately for alarm handling in the Alarm List and for event printout on printers.

Type 1 Alarms:	MV>H1	Type 2 Alarms:	MV>H2
	MV<L1		MV<L2
	DEV>H		
	DEV<L		

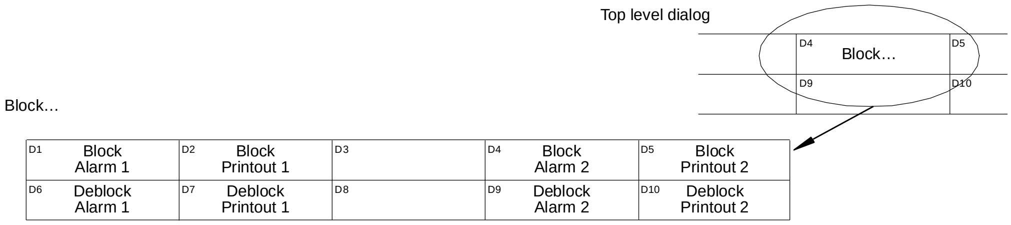


Figure 4-22. Dialog keys for alarm blocking

Beside this possibility for the operator to block alarms, blocking of alarms may be programmed as described in Section 3.1.13, Programmed Blocking of Limit Alarms.

4.4.6 Automatic Tuning

You access the dialog for automatic tuning via the top level dialog choice "Autotune…", which is available in the maintenance level. There are two ways of starting an autotuning:

- "Start Tune" causes a complete autotuning to be started.
- "Continue Tune" starts a shorter autotuning, skipping the first two phases.

Try to have the process in a steady state when you press "Start Tune". If the loop is being controlled with fairly large output movements before you start, try to make the start when the output value is near the middle of this variation. Otherwise the output value locked during the listen phase may cause an excessive offset in the measured value, and the output variations during the relay phase may not be sufficient for keeping the measured value around the desired level.

The alternative "Continue Tune" requires that an autotuning has been done before. You may want to retune due to changes in the process. Or you want to start from the present setting in another gain scheduling zone. Section 4.4.8 Gain Scheduling gives advice concerning the gain scheduling case.

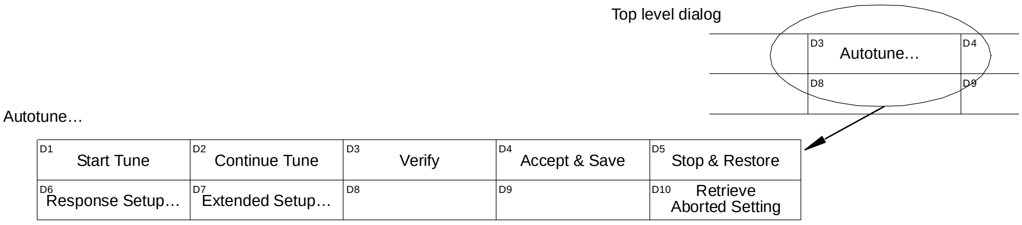


Figure 4-23. Dialog keys for autotuning

The tuning can be started from any of the modes MAN (but not Forced or Clamped), AUTO, E1, E2, or E3. You can also change mode during the autotuning, but "Man Forced" interrupts it. The mode indicated during the autotuning is the one that will be active when leaving autotuning. The mode also determines which setpoint source to use during the autotuning, as described in Section 4.1.7 Autotuning. If started in mode MAN or E3, where normally no setpoint value is used, the setpoint value to use in the autotuning is taken as the mean of the measured value obtained in the listen phase. If you want to do manual changes of the setpoint level used during the autotuning, you choose mode AUTO. Setpoint changes in the relay phase or in the verification phase will not be effective until the excited adaptive control phase is (re)entered (or the autotuning is left), but in the excited adaptive control phase they will have effect directly, i. e. in the next sample. If you want the loop to be in manual mode after the autotuning, you switch the mode back to MAN. Changing from another mode to MAN or E3 during autotuning does not change the setpoint value.

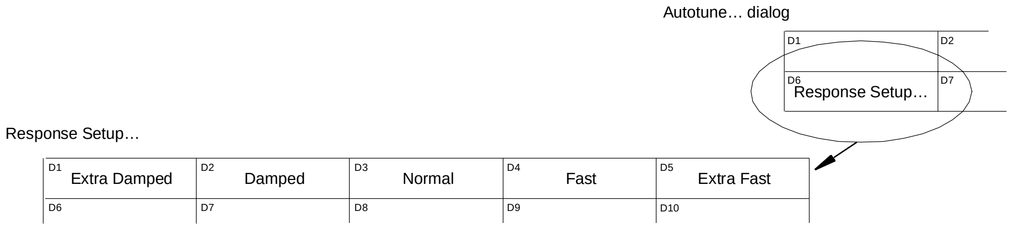


Figure 4-24. Dialog keys for selection of response type

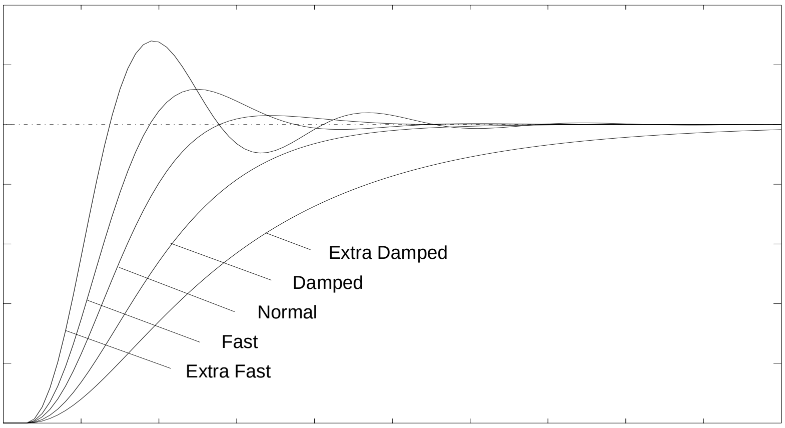


Figure 4-25. Typical step responses with different response type selected.

In the autotune dialog you have the key "Response Setup…" which enables you to specify the desired characteristics. In other words, what type of closed loop step response the autotuning should aim at:

- The default "Normal" means that you expect just a small overshoot. This alternative usually gives the quickest settling times.
- If you require that there shall be no overshoot at all, choose "Damped" or even "Extra Damped". The closed loop will then become, not only more damped, but also slower.
- If you can allow a bigger step response overshoot or even a higher degree of oscillations, and you require the loop to be as fast as possible in reaching a new setpoint or regaining the level after a load disturbance, choose "Fast" or even "Extra Fast". The closed loop will then be faster, but also more oscillative, than if you choose "Normal".

The following table gives approximate equivalents to other formulations of the aim for the tuning of the loop.

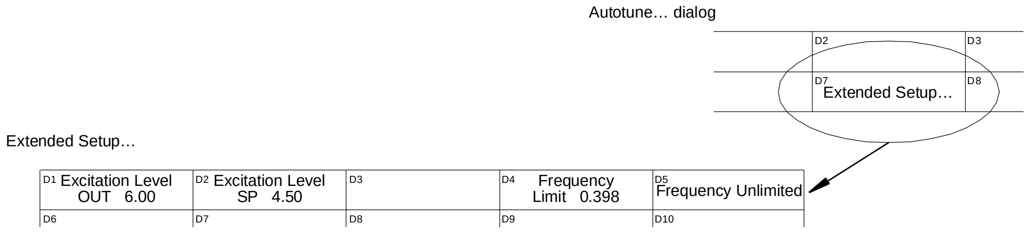
Table 4-8. Approximate equivalents to other formulations of the aim for the tuning of the loop

If you want	Choose response type
Overdamped	Extra Damped
Critically damped	Damped
ITAE optimal(1)	Normal
ISE optimal(2)	Fast or Extra Fast
Quarter amplitude damping	Extra Fast or Fast

(1) minimal Integrated Time-multiplied Absolute Error
(2) minimal Integrated Squared Error

A change of the response type will be taken into account immediately if autotuning or adaptive control is going on, and lead to new controller parameters as soon as the tuning algorithm has found a new accepted setting.

The autotuning aims at making the loop as fast as possible, while still giving the required type of response. In some cases it may give a too quick closed loop, in the sense that actuator limits or rate limits are reached too frequently, or actuator movements are too intense (causing wear). Should this happen, one can enter a "Frequency Limit" (under "Extended Setup…" in the autotune dialog), which limits how fast the automatic tuning is allowed to make the loop. Default is "Frequency Unlimited". To your guidance for selection of this frequency limit, the actual frequency associated with the present setting is shown in the object display, provided it was automatically tuned. In succeeding automatic tuning, this frequency will not be allowed to exceed the frequency limit you enter. For example, to make the closed loop half as quick, just enter a limit equal to half the presented frequency. Do not enter a too low limit, since that may require a negative gain and thus prevent the tuning method from finding any acceptable design.



Excitation Level OUT 6.00
Excitation Level SP 4.50
Frequency Limit 0.398
Frequency Unlimited

Figure 4-26. Dialog keys for excitation levels and design frequency limit

The excitation of the loop during the autotuning consists of output steps, due to a relay function feedback in the relay phase, and setpoint steps. The sizes of these steps (± excitation level) are selectable under "Extended Setup…" in the autotune dialog ("Excitation level OUT" and "Excitation level SP"). The default is 3% of their respective range. Entering the value 0 will cause a return to the default value. If the default excitation is considered too high (for a sensitive process loop), or too low (for a loop with much disturbances), you have to choose excitation levels that are suitable for the loop under consideration. Have in mind that a too low "Excitation level OUT" will not give proper relay oscillations.

In the calculations, it is assumed that output changes down to 0.1% give process responses in proportion to their size. In some cases you may know that this is not the case, for example if there is mechanical play or a deadzone in the actuator. That may fool the algorithms if unknown. You can therefore specify a deadzone value for the output (the key "DZ" under "Parameters…" in the top level dialog). Then, in the calculations, output changes smaller than DZ are not expected to give any process response at all. Thus, DZ is an adaptation deadzone for the output. Its exact value is normally non-critical. In case the three-point controller CON-PU1 is used for the actuator, its deadzone is also specified by this parameter, DZ.

Manual Intervention in the Autotuning

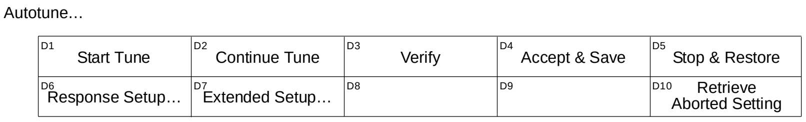
The progress of the autotuning can be followed in the object display, where the autotuning phase is shown together with a counter value for the number of steps. On successful completion, this text shows "Autotuning Done".

Sometimes you are alerted to make a decision. The question "Accept, Continue or Stop" appears in the object display. This is an alarm, "AT Quest", requiring acknowledgment. It remains active until you give one of the following three answers, in the "Autotune…" dialog.

- "Accept & Save" states that you consider the autotuning as finished. The result is saved.
- "Continue Tune" makes the autotuning continue by resuming the excited adaptive control phase.
- "Stop & Restore" interrupts the autotuning and discards the result. The previously saved parameter set is restored, if there was any.

These dialog choices are available also when not alerted for the decision. "Stop & Restore" can be applied at any time. "Continue Tune" only applies to the verification phase. "Accept & Save" requires that at least the excited adaptive control phase is reached. On entering one of these three choices, the "AT Quest" alarm is automatically acknowledged.

Another choice in the autotune dialog, that affects an ongoing autotuning, is "Verify". You use it if you want to force a verification test on the parameters presently used in the excited adaptive control phase, instead of waiting for it to enter the verification phase by itself.



Autotune...

D1 Start Tune	D2 Continue Tune	D3 Verify	D4 Accept & Save	D5 Stop & Restore
D6 Response Setup…	D7 Extended Setup…	D8	D9	D10 Retrieve Aborted Setting

Figure 4-27. The Autotune dialog

If any condition interrupts the autotuning before it is finished, or if it fails, the object display shows the text "Autotune OFF". If the reason was something else than operator commands ("Stop & Restore" or "Man Forced"), there is an alarm, and until it has been acknowledged the text "Failed" also appears in flashing red.

When autotuning is interrupted, the previously saved parameter set is always restored. But you may retrieve the discarded parameters, with "Retrieve aborted setting" in the autotune dialog, if the autotuning was aborted in the "Excite Phase" or the "Verify Phase". If the autotuning was aborted due to a condition that had nothing to do with the autotuning itself, it is often sensible to retrieve the setting that had been obtained half-way through the autotuning, and then continue it with "Continue Tune", instead of restarting the autotuning with "Start Tune".

It could also be meaningful in the case that verification failed at timeout, which means that the autotuning had trouble in either finding a good setting or just in verifying that it is good. In a fast process loop with a loaded Controller, the autotuning may have trouble in getting execution time for enough adaptation steps before timeout. If you suspect that it failed due to this, you should of course retrieve the setting and continue the tuning.

If autotuning was interrupted in the verify phase, and you do "Retrieve aborted setting", the parameters shown as "Present" will be the setting for which verification was tested. If you do "Save" and "Restore" before going on with "Continue Tune" (or adaptive control), you will start from this very setting. If you continue without first doing "Save" and "Restore", you will use a result of adaptation performed during that verification phase. This result has not been shown or used, but it is intended for use if the "Excite Phase" is resumed.

Tuning Setup Issues

The basic considerations for the choice of response type and (optionally) the speed for the closed loop were treated in Page 4-32 and Page 4-33 (Response Setup and Extended Setup dialogs). Some deeper considerations can be taken into account. One is the issue of robustness. Another is how the disturbance rejection is distributed over the frequency range. And yet another is how disturbances with different frequency content are causing actuator movement. The figures with Bode diagrams (gain versus frequency) below will be used to illustrate the effect of different choices on these properties of the closed loop.

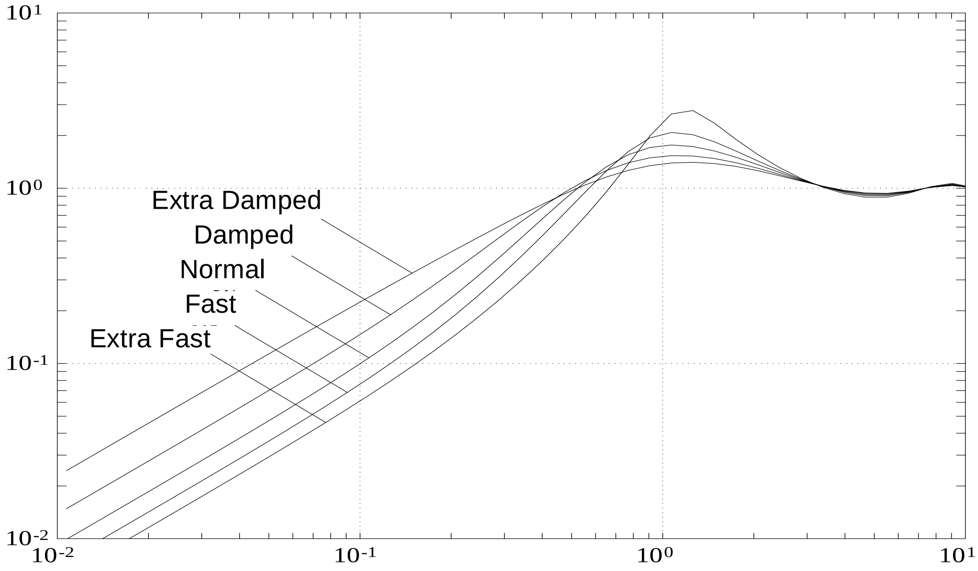


Figure 4-28. Sensitivity function for different choice of response type (unlimited frequency)

The higher the peak in the sensitivity function, the less robustness. So from Figure 4-28 you can conclude that if robustness is the main issue, one should choose Extra Damped. If further more robustness is required, it is obtained by reducing the speed of the closed loop, by choice of Frequency Limit. The peak of the sensitivity function (if there is any) normally occurs close to the characteristic frequency presented as result of the tuning.

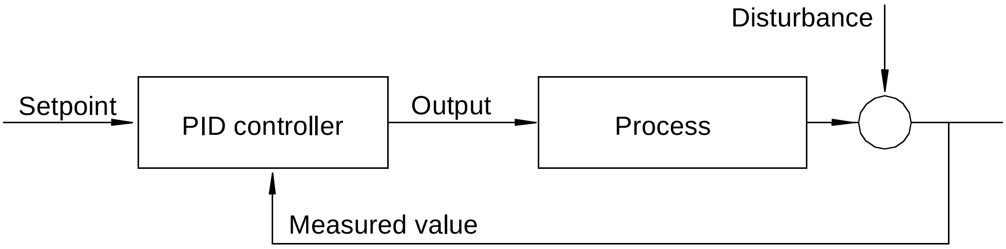


Figure 4-29. The sensitivity function is defined as the transfer function from disturbance to measured value in this block scheme

The disturbances in Figure 4-29 equal the measured value variations in open loop. They are modeled this way independent of what origin they actually have and where they in fact enter the loop. The measured value variations in closed loop (as caused by the disturbances) are obtained by multiplying the disturbance defined this way by the sensitivity function.

The sensitivity function thus illustrates how much disturbances in the measured value of different frequencies are reduced or amplified by the closed loop. Figure 4-28 is quite typical. High frequency disturbances can not be affected at all. Around the characteristic frequency they are amplified. And at low frequencies, they are reduced more the faster and less damped the closed loop is made. Lowering the closed loop speed in order to increase the robustness will have the positive side effect of reducing the amplification around the characteristic frequency, but a negative side effect of making the low frequency disturbance rejection less effective.

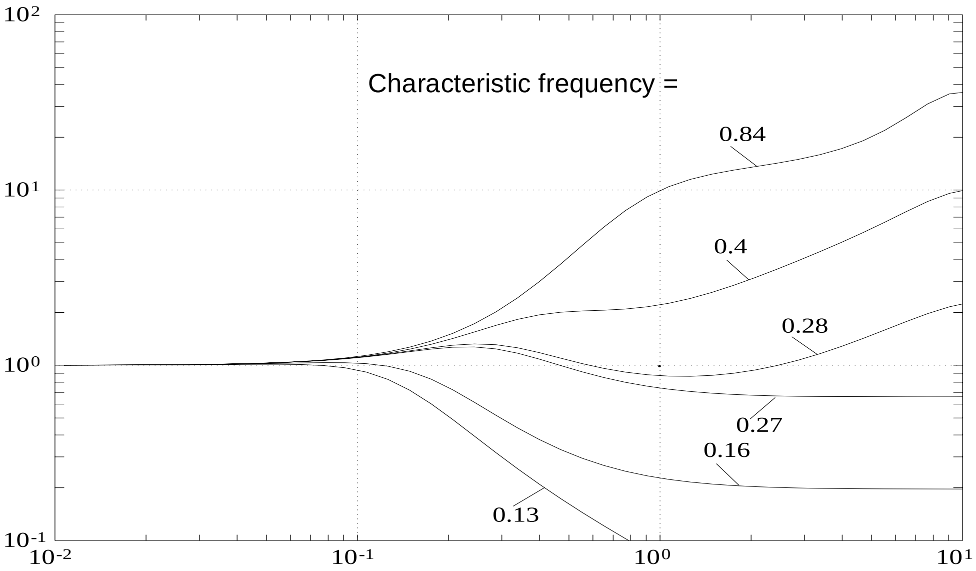


Figure 4-30. Gain from disturbance to controller output in closed loop

From the diagram in Figure 4-30 you can see how the gain from disturbance to controller output varies depending on the disturbance frequency for some different autotuned designs obtained with the same response type but different frequency limits. In the example shown, the fastest one (0.84) was obtained with No Limit chosen, and the rest of them were obtained with the respective frequency value chosen as Frequency Limit.

The three fastest ones resulted in PID control settings and the three slowest ones resulted in PI control. (Below 0.13 as limit would have required negative gain, so it was the limit for how slow the autotuning could make the loop with this choice of response type.) In cases with high frequency disturbances it is often advisable to skip derivative action and use only PI control. The way to obtain it with autotuning in PIDCONA is to reduce the frequency limit, if PID control was obtained at first.

4.4.7 Manual Tuning

We recommend you to use the built-in automatic tuning facility. But of course you have the possibility of entering a setting obtained otherwise, or to tune the controller manually. Via the top level dialog key "Parameters…", available in maintenance level, you can access the individual controller parameters, Gain, TI, TD, BETA, and TS, and the deadzone parameter DZ.

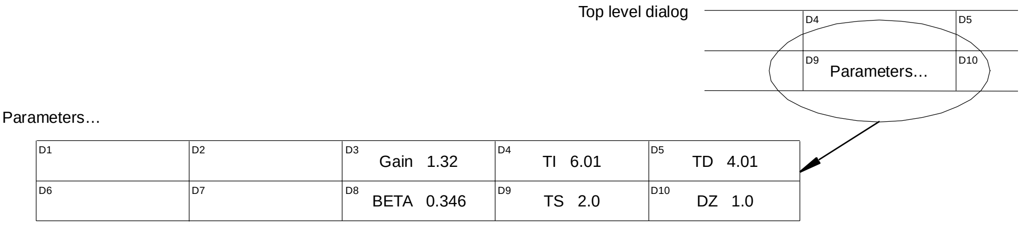


Figure 4-31. Dialog keys for manual setting of controller parameters and output deadzone

Gain, TI (integral time) and TD (derivative time) refer to the well-known parameters of a standard PID controller. TS is the sampling period. All times are expressed in seconds. BETA is the setpoint factor in the proportional part. (See further Section 4.1.2 Control Algorithm for the interpretation of these parameters.) Specifically, TI = 0 gives a controller without integral action, i. e. P or PD, and TD = 0 gives a controller without derivative action, i. e. P or PI.

If you change the value of any of the parameters, Gain, TI, TD, BETA, or TS, the setting will immediately be classified as a manual setting (as opposed to one that is autotuned or adapted). The indication (M) appears for the present setting in the parameters part of the object display. It means that there is no model associated with the present setting. Therefore it can neither be used in adaptive control, nor for starting a reduced autotuning with "Continue Tune". When the manual tuning is finished, the setting should be saved. This is done with the command "Save" described in Section 4.4.10 Save or Restore Tuned Parameters.

If you change the sampling period, TS, the new value is checked and possibly altered, since there are only twelve multiples of the configured cycle time to choose between. The nearest faster available multiple is chosen (or the fastest, if the entered sampling period is too short).

The output deadzone, DZ, specifies the deadzone for the three-point controller CON-PU1 when used for the actuator. It also specifies an adaptation deadzone for output variations in adaptive control, meaning that smaller variations than DZ are meaningless for the model updating and are therefore disregarded.

A Method for Improved Two-Degree-of-Freedom Control

The factor BETA allows two-degree-of-freedom control in PIDCONA. This means that the response to a setpoint step of a certain size differs from a disturbance step in the measured value of the same size. The autotuning often gives so high feedback gains (in order to reduce disturbances effectively), that it has to use a value of BETA less than one in order to avoid excessive overshoots in setpoint step responses. This is the normal case, obtained automatically. But this factor can also be used to increase the setpoint gain (thereby speeding up the setpoint step response) in cases where the feedback gains had to be kept low due to a high disturbance level. A method to do this by first autotuning and then adjusting the result is described below.

1. Specify Extra Damped in the Response Setup menu, and start the autotuning.
2. When you see roughly what frequency value the tuning seems to settle at during the autotuning, set the Frequency Limit to roughly half that value (in the Extended Setup menu)
3. Let the autotuning finish, or choose Accept & Save to finish it when you have seen that it produces a calm control with small actuator movements in spite of the noisy measurement value.
4. Increase the obtained value of BETA by a factor 10 manually.
5. Check the setpoint step response, and save the setting by the choice Accept & Save in the Autotuning menu or Save in the Adaptation menu.

The result is a closed loop that responds quickly to setpoint changes but slowly to variations in the measurement value. This is ideal for the inner loop in a cascade control, if you can't make the feedback fast due to high disturbance level, but you need fast response to the outputs from the outer loop controller for the sake of the overall control.

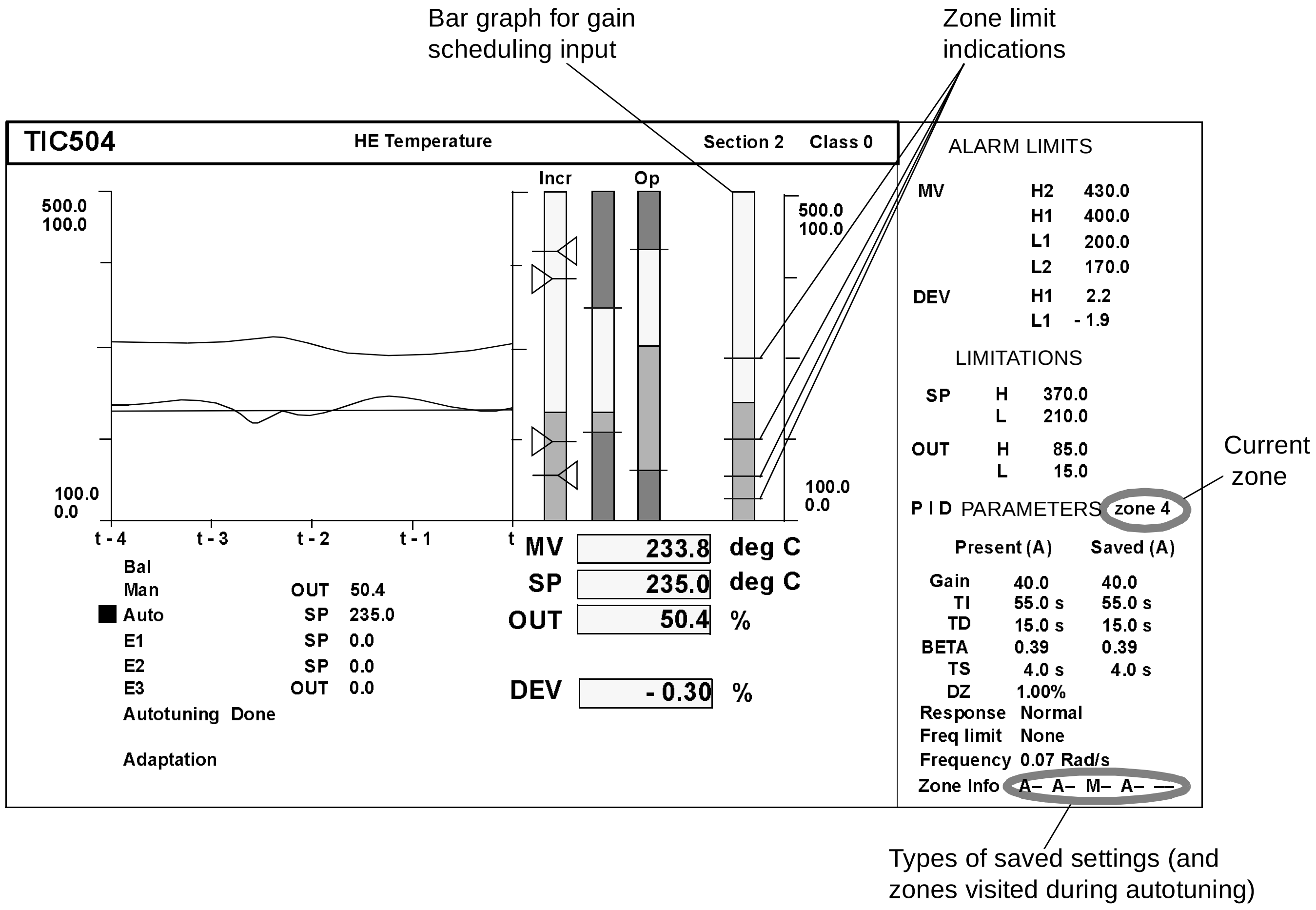
4.4.8 Gain Scheduling

Object Display Information

If the control loop is configured for gain scheduling (C6 included in the PC element), there is a scheme with five zones, for five different controller settings. Then the object display has a bar graph for the scheduling variable, and a line "Zone Info" in the "Parameters" area. Gain scheduling information is only shown in the object display, and not in the other display elements.

The bar graph also shows the zone limits, so that you can see where among the zones the scheduling input value is at present. While gain scheduling is enabled, the active zone number is shown in the "Parameters" area, and the "Saved" parameters presented are those belonging to this zone. While gain scheduling isn't enabled, the "Saved" parameters always means those of zone 1.

The "Zone Info" line shows in which zones the controller has been tuned, i. e. in which zones a setting has been saved. A manually tuned setting is indicated by an "M" and an automatically tuned setting by an "A", from left to right for zones 1 through 5. Untuned zones are marked "-".



Bar graph for gain scheduling input

Zone limit indications

Current zone

Types of saved settings (and zones visited during autotuning)

TIC504 HE Temperature Section 2 Class 0

Bal
Man OUT 50.4
■ Auto SP 235.0
E1 SP 0.0
E2 SP 0.0
E3 OUT 0.0
Autotuning Done

Adaptation

MV 233.8 deg C
SP 235.0 deg C
OUT 50.4 %
DEV - 0.30 %

ALARM LIMITS

MV	H2	430.0
	H1	400.0
	L1	200.0
	L2	170.0
DEV	H1	2.2
	L1	- 1.9

LIMITATIONS

SP	H	370.0
	L	210.0
OUT	H	85.0
	L	15.0

P I D PARAMETERS zone 4

	Present (A)	Saved (A)
Gain	40.0	40.0
TI	55.0 s	55.0 s
TD	15.0 s	15.0 s
BETA	0.39	0.39
TS	4.0 s	4.0 s
DZ	1.00%	
Response	Normal	
Freq limit	None	
Frequency	0.07 Rad/s	
Zone Info	A– A– M– A– ––	

Figure 4-32. A PIDCONA object display with gain scheduling information

Tuning

The parameter scheme is readily built by autotuning once in each zone. A setting obtained for one zone is suitable to use as a starting point for another zone, if the time scale of the process is reasonably similar in the two zones. When entering a zone without a saved setting, the present setting remains unchanged. It is thus available for a start of autotuning with "Continue Tune". It also means that, as long as only one zone has been tuned, that zone's setting will be used in all zones. The procedure can be summarized as follows.

1. Autotune in one zone, using "Start Tune"
2. Move to the next untuned zone
3. Optionally, press "Save" followed by "Restore" in this zone (explained below)
4. Autotune in this zone with "Continue Tune"
5. Repeat steps 2 - 4 until all desired zones have been tuned.

The "Save" and "Restore" of step 3 are not compulsory. "Save" means that the setting from the previous zone is saved in the new zone. Then, if the autotuning in the new zone fails or is interrupted, that setting remains there. Otherwise there would still be no setting saved there after interruption. "Restore" causes a higher initial adaptation rate. It also ensures that the sampling rate choice is centered around the obtained sampling rate of the previous zone.

(When using "Start Tune", the set of sampling rates to choose among is determined by the relay phase. The set remains the same until either an automatically tuned saved setting is restored, or a new autotuning is in the relay phase.)

It was assumed above that the present setting is retained, while SCHEDIN passes from the zone having the desired start setting to the new zone to be tuned. This is the case if, either there is no previous setting stored in the new zone, or the gain scheduling is disabled when SCHEDIN enters the new zone. In the latter case gain scheduling must be re-enabled before starting the tuning, unless the new setting is intended for zone 1. Enabling and disabling is done via the PC element input GSENBL.

The setting obtained in an autotuning session with gain scheduling enabled belongs to the zone where it was started. The active zone remains the same throughout the whole autotuning, even if the scheduling variable crosses zone limits. In the object display, you get notified by the text "Outside start zone" if this happens in the verification phase. Also a "V" is indicated in the "Zone Info" line for those zones, other than the active one, which have been visited by the scheduling variable during the on-going autotuning. The normal zone limit hystereses apply.

4.4.9 Adaptive Control

The top level dialog choice "Adaptation…", available in the maintenance level, opens an adaptation dialog. If the control loop has been autotuned, the choice "Adaptation Enable" enables adaptive control. To get adaptive control there are further conditions that must be fulfilled. There may, for example, be conditions programmed in AMPL that take into account the present state of operations in the process, and switch the adaptation on and off while it is enabled by the operator. See also Section 4.1.9 Adaptive Control. The operator choice "Adaptation Off" switches it off unconditionally.

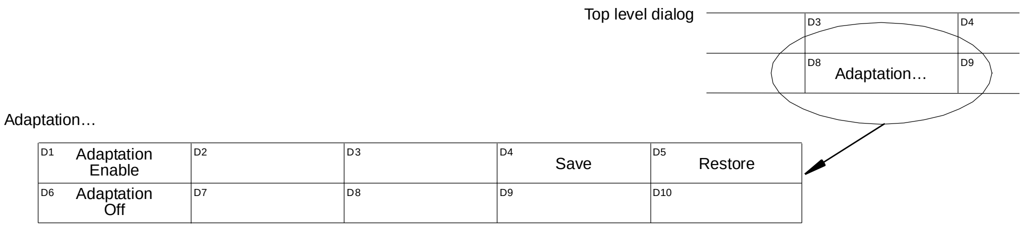


Figure 4-33. Dialog keys concerning adaptive control and saved settings

The display elements that show control mode also show whether the control is adaptive or not. The adaptation can only be active in automatic modes (AUTO, E1, or E2). The object display shows a green box before the text "Adaptation" during adaptive control, a yellow box when adaptation is enabled but not active, and otherwise no box. The other display elements show "Ad" together with the control mode during adaptive control. They give no indication, when it is enabled but not active.

4.4.10 Save or Restore Tuned Parameters

In the adaptation dialog, there are the dynamic keys "Save" and "Restore":

- "Save" saves the present setting, thereby replacing the one previously saved (the one in the active zone if gain scheduling is enabled)
- "Restore" restores the saved setting, thereby replacing the present setting

The present setting to be saved can be of any type: autotuned, manually tuned, obtained during adaptive control, or classified as "no setting" (i. e. with Gain = 0). Saving "no setting" means that you delete the previous saved setting, if any.

In the object display, both the present and the saved settings are shown. If there is no setting saved in the current zone, it can of course not be restored.

4.5 Runtime Operation Menus

The dialogs used for manipulating the object are described in Section 4.4 Operating Instructions and in Appendix B, Process Dialog.

Chapter 5 Maintenance

5.1 Preventive Maintenance

Maintenance is not really applicable to the PIDCONA function, except possibly making sure that it stays tuned. It is not designed to give any message to the user when retuning is needed. If the process dynamics stays unchanged this will never be needed, if it was well tuned initially. However, if tight deviation alarm limits are used and deviation alarms tend to become more frequent, this fact may be taken as a hint that perhaps the process dynamics has changed so that the controller needs to be retuned.

If the loop was autotuned before and you decide that retuning is needed, you simply do a new autotuning. Unless you have reason to believe that there has been a large change, you choose the shorter version commanded by "Continue Tune". It will then start operating with adaptive control and setpoint excitation and end with a new verified setting. Retuning this way can often be done without upsetting the normal production in the process. You just have to find a suitable compromise, between low enough setpoint excitation level to keep the variations acceptable, and high enough setpoint excitation level to produce relevant data for the adaptation (high enough signal-to-noise ratio).

5.2 Hardware Indicators

There are no hardware indicators specific for the functional unit PIDCONA.

5.3 Error Messages

Alarms may be taken as an indication that something is not operating normally. Compare Section 4.4.4, Alarms where the alarms are listed, and Section 5.4 Fault Finding and User Repair where causes are explained and counteractions are suggested.

A pair of system messages (error codes 101 and 100) occur the first time the Controller is taken out of configuration mode (with DICONFIG), if there is too little memory for the dimensioned number of PIDCONA instances (in DIMDB). See Section 3.3, Application Start-up.

5.4 Fault Finding and User Repair

This section aims at assisting you in finding the causes when you get into trouble, and to give hints about suitable counteractions when that is appropriate.

5.4.1 Invalid Sampling Rate

If you have a "TS Fault" alarm, it was probably caused by a change of cycle time in the control module. The result was that there is, in at least one of the gain scheduling zones, a saved sampling period, TS, that can not be achieved using the present control module cycle time. The alarm will remain until all zones either have nothing saved or have a saved setting with a sampling period that can be achieved.

To get rid of the "TS Fault" alarm, change the control module cycle time back, or save new settings in the zones where the sampling period has become invalid. Get the new settings by autotuning so that you can use the benefits that it gives. Otherwise you can just change TS.

5.4.2 Fault Finding Regarding Autotuning

If you try to use autotuning and it does not work, or does not produce the type of result you expected, the list below may assist you in finding the reason and possible counteractions.

Table 5-1. Troubleshooting List

Autotuning Problem	Cause and Counteraction
There are no output oscillations in the relay phase	The autotuning was started at an unfortunate moment, locking the output with its excitation to levels which are not in the range required for making the measured value cross the setpoint value. Try to start the autotuning when the process is in a steady state. Do it in mode MAN or with the setpoint roughly equal to the measured value. If in mode AUTO, with oscillations, start when the output has a middle value.
	A disturbance has moved the level of operation, so that the output is required to move further than the excitation level selected allows it to do. The excitation level is not high enough to make the measured value cross the setpoint value. Start a new autotuning, possibly with a higher excitation level for the output.
	The function parameter REVACT of the PC element is wrong. With REVACT=1, a positive response is expected to a positive output step. With REVACT=0, a positive step is expected to give a negative response. With the wrong value of REVACT, there will be no oscillations. (An initial response in the opposite direction is no cause to change it, provided the steady state response has the right sign.) Correct REVACT and start a new autotuning.

Table 5-1. Troubleshooting List

Autotuning Problem	**Cause and Counteraction**
The oscillations in the relay phase are very slow	The process itself is very slow. It takes this long time for it to react to the changes of the output. Then the behaviour is correct. You just need more patience.
	There was big variations in the measured value when starting the autotuning. Therefore a large value is used for the relay hysteresis, which leads to slower oscillations. If these big variations are not representative for the normal steady state behaviour, try a new start of the autotuning at an instant in steady state with less disturbances.
	The excitation level for the output is low, in combination with a significant relay hysteresis. The response to the output changes barely exceeds the hysteresis. Increase the excitation level for the output and start a new autotuning.
The verification repeatedly fails to accept the setting	The rate limit SPEEDO (the actuator speed) prevents the response from being as quick as the design frequency says. If you are satisfied with the responses produced, finish the autotuning by pressing "Accept & Save". Increase SPEEDO, if it was chosen smaller than the available actuator speed. Set a frequency limit to make the design slower. Decrease the excitation level for the setpoint to reduce the effect of the rate limit.
	The disturbance level is higher than estimated. If you are satisfied with the responses produced, finish the autotuning by pressing "Accept & Save".
The obtained control is too slow	Due to the disturbance pattern, or an unstationary state of the process at start of the autotuning, the relay oscillations became too slow, and thus the speed of the process was underestimated. Make a new autotuning with "Start Tune".
	The design frequency was limited by the frequency limit specified. Increase the frequency limit, or remove it. Default is "Frequency Unlimited".
	The specified response type is too damped. Alter the response type in "Response Setup" towards less damped or faster.
	The sampling period became equal to the configured cycle time for the control module, but it is still too long. Decrease the cycle time for the control module.

Table 5-1. Troubleshooting List

Autotuning Problem	Cause and Counteraction
The obtained control gives too large overshoots or oscillations	The specified response type is too fast. Alter the response type in "Response Setup" towards slower or more damped.
The obtained control seems nervous, with persisting rapid output movements	The control reacts too much on high frequency disturbances. Set a frequency limit (in the "Extended Setup" dialog), lower than the present frequency and continue the tuning or start a new one, to make the design slower. Apply a suitable low pass filter to the measured value before its connection to the PC element input MV, to reduce its content of high frequency disturbances. Then make a new tuning with "Start Tune" or "Continue Tune".

5.4.3 Fault Finding regarding Adaptive Control

Alarms

An alarm "Ad Fail" may occur, while using adaptive control. It means that, during some time, the adaptation calculations did not succeed in finding settings that it accepts. There may be several reasons for this. Suppose adaptive control is normally used without getting this alarm. Then some possible reasons for the alarm to appear are the following:

- The process has changed in a way that PIDCONA could not adapt to, or that it requires longer time to adapt to. Maybe this is due to a fault in the process, sensors, or actuators.
- The disturbance level has increased so much that the consequently higher modeling uncertainty makes it very difficult to find designs that are considered robust.
- The model has become bad due to adaptation using irrelevant data, e. g. with output variations less than the actuator deadzone, where the deadzone, DZ, is specified too small.
- You have changed specifications for the controller. If you change the type of response or decrease the frequency limit, it is not always possible to find an accepted design for the new values.

If there are frequent "Ad Fail" alarms, it is probably not suitable to use adaptive control for the loop. At least not in the circumstances where the alarms occur. If adaptive control is still needed in other situations, you should add conditions to the PC element input ADAPON, or use gain scheduling where the setting used during the particular circumstances is a manual one (thereby avoiding adaptation).

No Adaptation Obtained

Possible causes why the parameters may remain constant although adaptive control is indicated:

- There is no change of the process. (Often small variations are seen in this case.)
- The signal variations are lower than the size of the deadzones.
- There is no acceptable new setting found. Therefore the old one remains.
- The Controller's processing load is so high that virtually no capacity is left for adaptation.

The main idea of adaptation deadzones is to avoid using irrelevant data for adaptation, when the control actions do not provide enough excitation to get any valid information about the process. If you have adaptive control but do not see any parameter variations for a long time, it may be due to signal variations lower than the size of the deadzones. If this happens in situations where there ought to be adaptation, the deadzones may be too large.

The adaptation deadzone for the output, specified by DZ, is described in Section 4.4.7, Manual Tuning. It is specified by the user.

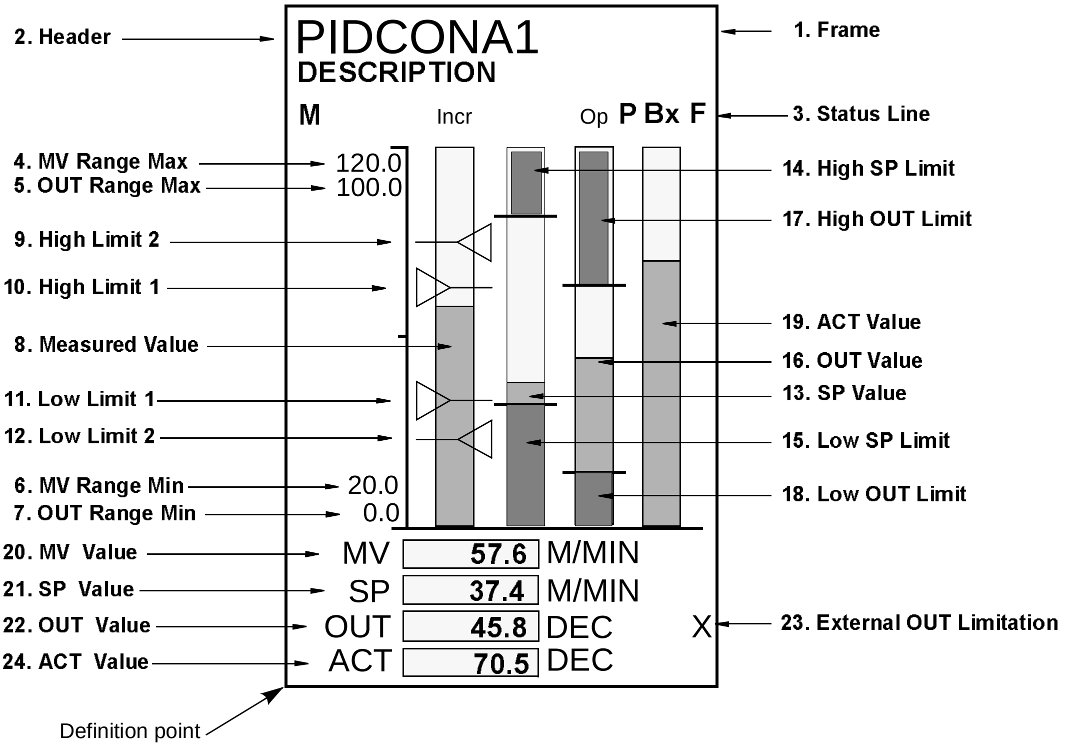
There is also a similar adaptation deadzone for the measured value. Its value is determined automatically during the autotuning. There is one value for each gain scheduling zone. They are accessible in the subordinate data base element "PID adaptation data", part "Auxiliary model parameters". It is possible to change the values manually. Use this possibility only if there are strong reasons for believing that the obtained value is unsuitable. After a change, the new value is not used until a "Restore" is made.

Appendix A Display Elements

A.1 General

This appendix describes the display elements available for PIDCONA in the Operator Station. The descriptions are reprinted from *AdvaCommand User Interface Reference Manual*.

A.1.1 Display Element for Group display: PIDAGR01



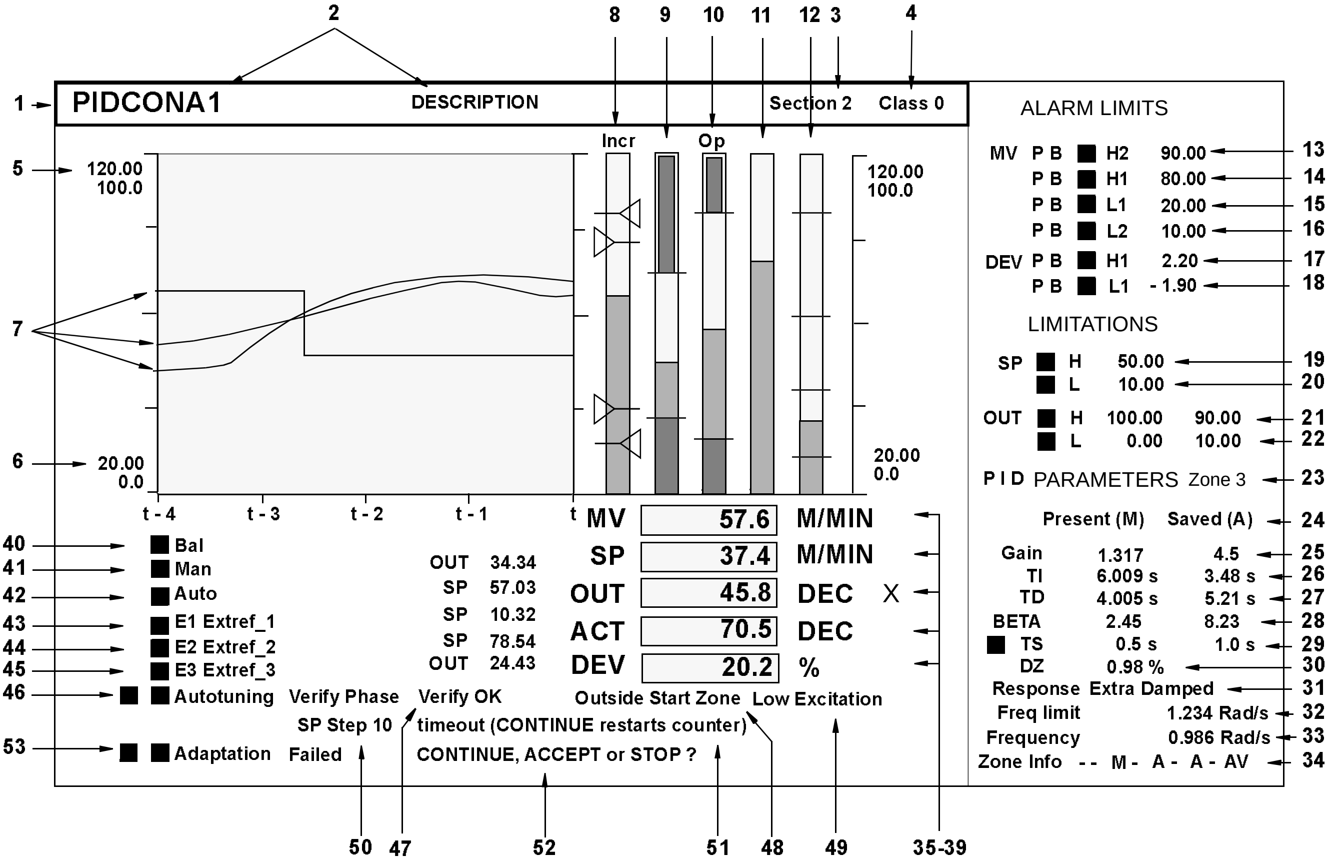
No	Description	Remarks
1	**SelectFrame**	Selected
2	**Header**	Object name and description
3	**Status Line**	Status Line
3.1	**Mode**	
3.2	**Direction**	
3.3	Opening or Closing	

No	Description	Remarks
16	**OUT Value**	Output value
17	**High OUT Limit**	Upper output limit
18	**Low OUT Limit**	Lower output limit
19	ACT Value	Actuator value
20	**Measured Value**	The MV object value
20.1	**MV**	

No	Description	Remarks
3.4	Print Block	
3.5	Alarm Block	
3.6	Alarm	Alarm
4	**MV Range Max**	Max Range of the MV and SP values
5	**OUT Range Max**	Max Range of the OUT and ACT values
6	**MV Range Min**	Min Range of the MV and SP values
7	**OUT Range Min**	Min Range of the OUT and ACT values
8	**Measured Value Bargraph**	
9	**High Limit 2**	Alarm High limit 2
10	**High Limit 1**	Warning High limit 1
11	**Low Limit 1**	Warning Low limit 1
12	**Low Limit 2**	Alarm Low limit 2
13	**SP Value**	Setpoint value
14	**High SP Limit**	Upper setpoint limit
15	**Low SP Limit**	Lower setpoint limit

No	Description	Remarks
20.2	**MV Value**	
20.3	**MV Unit**	
21	**Setpoint Value**	The SP object value
21.1	**SP**	
21.2	**SP Value**	
21.3	**SP Unit**	
22	**Output Value**	The OUT object value
22.1	**OUT**	
22.2	**OUT Value**	
22.3	**OUT Unit**	
23	External OUT Limitation	OUT limited by external limitation
24	**Actuator Value**	The ACT object value
24.1	**ACT**	
24.2	**ACT Value**	
24.3	**ACT Unit**	

A.1.2 Object Display: PIDAOB01



PIDCONA1 DESCRIPTION Section 2 Class 0

120.00
100.0

20.00
0.0

120.00
100.0

20.00
0.0

MV 57.6 M/MIN
SP 37.4 M/MIN
OUT 45.8 DEC X
ACT 70.5 DEC
DEV 20.2 %

■ Bal
■ Man
■ Auto
■ E1 Extref_1
■ E2 Extref_2
■ E3 Extref_3
■ ■ Autotuning Verify Phase Verify OK Outside Start Zone Low Excitation
SP Step 10 timeout (CONTINUE restarts counter)
■ ■ Adaptation Failed CONTINUE, ACCEPT or STOP ?

OUT 34.34
SP 57.03
SP 10.32
SP 78.54
OUT 24.43

ALARM LIMITS

MV P B ■ H2 90.00
P B ■ H1 80.00
P B ■ L1 20.00
P B ■ L2 10.00
DEV P B ■ H1 2.20
P B ■ L1 - 1.90

LIMITATIONS

SP ■ H 50.00
■ L 10.00
OUT ■ H 100.00 90.00
■ L 0.00 10.00

P I D PARAMETERS Zone 3

	Present (M)	Saved (A)
Gain	1.317	4.5
TI	6.009 s	3.48 s
TD	4.005 s	5.21 s
BETA	2.45	8.23
■ TS	0.5 s	1.0 s
DZ	0.98 %	

Response Extra Damped
Freq limit 1.234 Rad/s
Frequency 0.986 Rad/s
Zone Info - - M - A - A - AV

No	Description	Remarks
1	SelectFrame	Object select frame
2	Header	Object name and description
3	Section	Text in front of the value
4	Class	Text in front of the value
5	MV Range Max	Range Max of MV and SP values
	OUT Range Max	Range Max of OUT and ACT values
6	MV Range Min	Range Min of MV and SP values
	OUT Range Min	Range Min of OUT and ACT values
26.4	Unit	
26.5	Saved TI Value	
26.5	Unit	
27.1	TD Text	Derivation time
27.2	Present TD Value	
27.3	Unit	
27.4	Saved TD Value	
27.5	Unit	

No	Description	Remarks
7	Measured Value Trim Curve	
8.1	Direction	Increase. When measured value is increasing, it is counteracted with decreasing output.
8.2	Measured Value Bargraph	Measured Value
8.3	High Limit 2	Alarm High limit 2
8.4	High Limit 1	Warning High limit 1
8.5	Low Limit 1	Warning Low limit 1
8.6	Low Limit 2	Alarm Low limit 2
9.1	SP Value Bargraph	Setpoint Value
9.2	High SP Limit	Upper limit setpoint
9.3	Low SP Limit	Lower setpoint limit
10.1	Opening or Closing	Opening. When output is increasing the valve is opening.
10.2	OUT Value Bargraph	Output value
10.3	HIgh OUT Limit	Upper limit output
10.4	Low OUT Limit	Lower limit output
11	ACT Value Bargraph	Actuator position.
12.1	Gain Scheduling Bargraph	
12.2	Zone limit between zone 4 and 5	
12.3	Zone limit between zone 3 and 4	
12.4	Zone limit between zone 2 and 3	
12.5	Zone limit between zone 1 and 2	
13	ALARM LIMITS	Alarm limits
13.1	MV Text	Text in front of the line
13.2	Printout blocked	Printout blocked.
13.3	Alarm blocked	
13.4	Alarm Indication	
13.5	H2 Text	Text in front of the value
13.6	H2 Value	High limit 2 value

No	Description	Remarks
28.1	BETA Text	Setpoint factor BETA
28.2	Present BETA Value	
28.3	Saved BETA Value	
29.1	Alarm Indication	
29.2	TS Text	Sampling time
29.3	Present TS Value	
29.4	Unit	
29.5	Saved TS Value	
29.6	Unit	
30.1	DZ Text	Deadzone
30.2	Value	
30.3	Unit	
31.1	Response Text	Response type
31.2	Response Type Text	
32.1	Freq Limit Text	Design frequency limit
32.2	Value	
32.3	Unit	
33.1	Frequency Text	Design frequency. Shown only when an autotuned setting exist.
33.2	Value	
33.3	Unit	Design frequency unit.
34.1	Zone Info Text	Gain scheduling zone information
34.2	Text	
34.3	Text	
34.4	Text	
34.5	Text	
34.6	Text	
35.1	MV Text	Text in front of the value

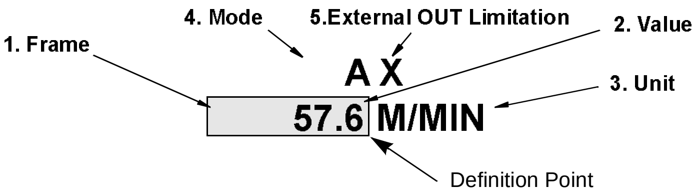
No	Description	Remarks
14.1	Printout blocked	
14.2	Alarm blocked	
14.3	Alarm Indication	
14.4	H1 Text	Text in front of the value
14.5	H1 Value	High limit 1 value
15.1	Printout blocked	
15.2	Alarm blocked	
15.3	Alarm Indication	
15.4	L1 Text	Text in front of the value
15.5	L1 Value	Low limit 1 value
16.1	Printout blocked	
16.2	Alarm blocked	
16.3	Alarm Indication	
16.4	L2 Text	Text in front of the value
16.5	L2 Value	Low limit 2 value
17.1	DEV	
17.2	Printout blocked	
17.3	Alarm blocked	
17.4	Alarm Indication	
17.5	H Text	Text in front of the value
17.6	Value	
18.1	Printout blocked	
18.2	Alarm blocked	
18.3	Alarm Indication	
18.4	L Text	Text in front of the value
18.5	L Value	
19	LIMITATIONS	Limitations
19.1	SP Text	Setpoint
19.2	Warning Indication	Setpoint is equal to high setpoint limit

No	Description	Remarks
35.2	MV Value	Signal error
35.3	MV Unit	Measured unit
36.1	SP Text	Text in front of the value
36.2	SP Value	Setpoint value
36.3	SP Unit	Setpoint unit
37.1	OUT Text	Text in front of the value
37.2	OUT Value	Output value
37.3	OUT Unit	Output unit
37.4	External OUT Limitation Indication	OUT limited by external limitation
38.1	ACT Text	Actuator position
38.2	ACT Value	Actuator value
38.2	ACT Unit	Actuator unit
39.1	DEV Text	Text in front of the value
39.2	DEV Value	Deviation value. DEV = MV - SP
39.3	DEV Unit	Deviation unit
40.1	Bal Indicator	Balance control mode
40.2	Bal Mode	
40.3		Balance Local control mode
41.1	Man Indicator	When autotuning is finished manual control mode is resumed
41.2	Man Mode	
41.3		Man Clamped control mode
41.4	Out Text	Text in front of the value
41.5	Man Out Value	Man Out value
42.1	Auto Indicator	When autotuning is finished control mode auto is resumed
42.2	Auto Mode	Auto control mode
42.3	SP Text	Text in front of the value
42.4	Auto Setpoint Value	Auto Setpoint value
43.1	E1 Indicator	When autotuning is finished control mode E1 is resumed
43.2	E1 Mode	Text behind the indication

No	Description	Remarks
19.3	H Text	High limit setpoint
19.4	H Value	
20.1	Warning Indication	Setpoint is equal to low setpoint limit
20.2	L Text	Low limit setpoint
20.3	L Value	
21.1	OUT Text	Output
21.2	Warning Indication	Output is equal to output high limit
21.3	H Text	Output high limit.
21.4	H Value	Output high limit value.
21.5	External H value	External output high limit value
22.1	Warning Indication	Output is equal to low output limit
22.2	L Text	Output low limit.
22.3	L Value	Output low limit value.
22.4	External L Value	External output low limit value.
23	PARAMETERS	
23.1	P	Gain active
23.2	I	Integration active
23.3	D	Derivation active
23.4	Zone Text	Gain scheduling active
23.5	Zone Value	Gain scheduling zone
24.1	Present Text	Present parameters
24.2	Text	Default setting. Invisible.
24.3	Saved Text	Saved parameters
24.4	Text	Default setting. Invisible.
25.1	Gain Text	Gain
25.2	Present Gain Value	
25.3	Saved Gain Value	
26.1	TI Text	Integration time
26.2	Present TI Value	

No	Description	Remarks
43.3	E1 Name	User text behind the text E1
43.4	SP Text	Text in front of the value
43.5	E1 Setpoint Value	E1 control mode setpoint
44.1	E1 Mode Indicator	When autotuning is finished control mode E2 is resumed
44.2	E2 Mode	E2 control mode
44.3	E2 Name	
44.4	SP Text	Text in front of the value
44.5	E2 Setpoint Value	E2 control mode setpoint
45.1	E3 Mode Indicator	When autotuning is finished control mode E3 is resumed
45.2	E3 Mode	External 3 control mode
45.3	E3 Name	
45.4	OUT Text	Text in front of the value
45.5	E3 Output Value	E3 control mode output
46.1	Autotuning Alarm Indicator	Unacknowledged alarm
46.2	Autotuning Indicator	Autotuning active
46.3	Autotuning	Autotuning active.
46.4	Text	
47	Text	
48	Text	
49	Text	
50.1	Text	
50.2	Counter	Counter value.
51	Text	
52	Text	
53.1	Adaptation Alarm Indicator	
53.2	Adaptation Indicator	Adaptation active.
53.3	Adaptation	Adaptation active.
53.4	Text	Alarm text.

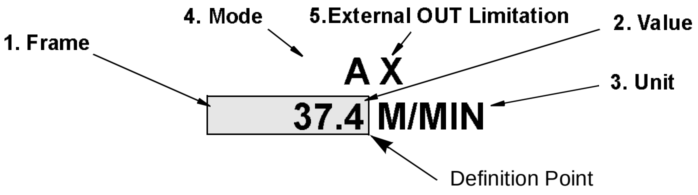
A.1.3 Display Element for Process display: PIDAPD01



No	Description	Remarks
1	SelectFrame	Selected
2	Measured Value	Object value
3	Unit	Object unit

No	Description	Remarks
4	Mode	Object mode
5	External OUT Limitation	External output limitation

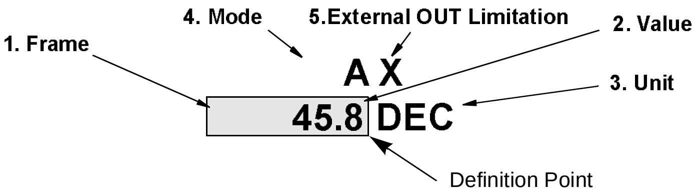
A.1.4 Display Element for Process display: PIDAPD02



No	Description	Remarks
1	SelectFrame	
2	SP Value	Setpoint value
3	Unit	Object unit

No	Description	Remarks
4	Mode	Object mode
5	External OUT Limitation	External output limitation

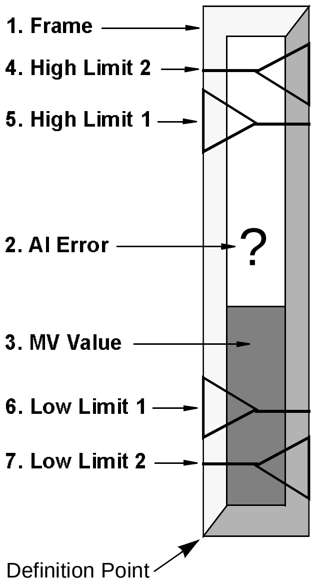
A.1.5 Display Element for Process display: PIDAPD03



No	Description	Remarks
1	SelectFrame	
2	Output Value	Output value
3	Unit	Object output unit

No	Description	Remarks
4	Mode	Object mode
5	External OUT Limitation	External output limitation

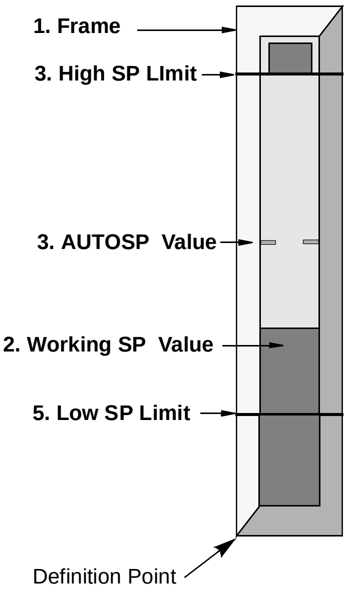
A.1.6 Display Element for Process display: PIDAPD51



No	Description	Remarks
1.1	SelectFrame	Object frame
1.2	TopLeftEdge	Three Dimensions effect TopLeftColor
1.3	BottomRightEdge	Three Dimensions effect BottomRightColor
2	AI Error	Unacknowledged signal error
3	MV Value	Object value

No	Description	Remarks
3	MV Value	Object value
4	High Limit 2	Alarm High limit 2
5	High Limit 1	Warning High limit 1
6	Low Limit 1	Warning Low limit 1
7	Low Limit 2	Alarm Low limit 2

A.1.7 Display Element for Process display: PIDAPD52

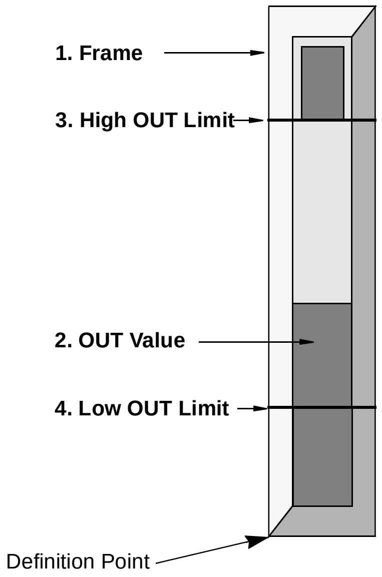


No	Description	Remarks
1.1	SelectFrame	Object frame
1.2	TopLeftEdge	Three Dimensions effect TopLeftColor
1.3	BottomRightEdge	Three Dimensions effect BottomRightColor

No	Description	Remarks
2	Setpoint Value	Object setpoint value
3	High SP Limit	Upper setpoint limit
4	Low SP Limit	Lower setpoint limit

A.1.8 Display Element for Process display: PIDAPD53

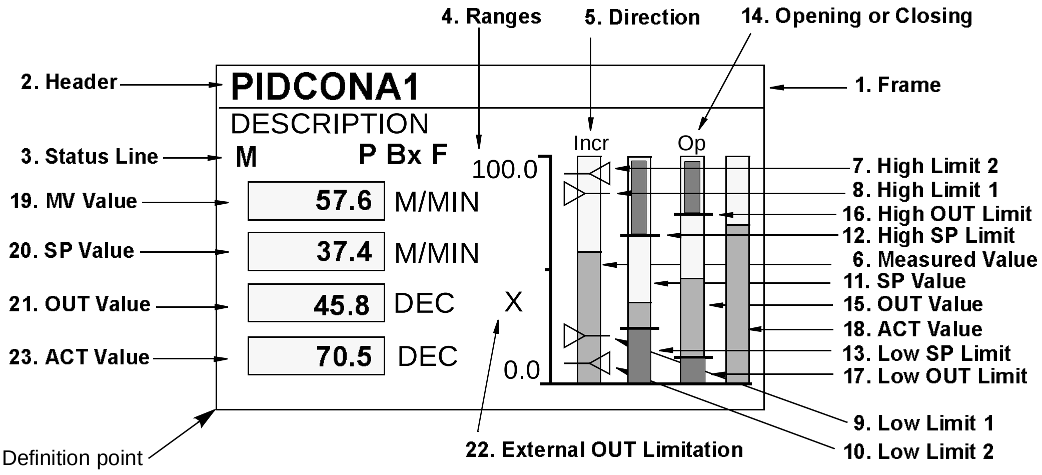
:



No	Description	Remarks
2	Output Value	Output value
3	High OUT Limit	Upper output limit
4	Low OUT Limit	Lower output limit

No	Description	Remarks
1.2	TopLeftEdge	T
1.3	BottomRightEdge	
2	Output Value	Output value

A.1.9 Display Element for Object close-up display: PIDAWI01



No	Description	Remarks
1	SelectFrame	Object select frame
2	Header	Object name and description
3	Status Line	
3.1	Mode	Bal Local mode
3.2	Printout Blocked	Printout of alarm blocked
3.3	Alarm Blocked	Alarm blocked by operator
3.4	Alarm	Alarm
4.1	Range Max	Range Max of the MV and SP values
4.2	Range Min	Range Min of the MV and SP values
5	Direction	Increase. When measured value is increasing, it is counteracted by with increasing output.
6	Measured Value	Measured value
7	High Limit 2	High Alarm limit 2
8	High Limit 1	High Warning limit 1
9	Low Limit 1	Low Warning limit 1

No	Description	Remarks
14	Opening or Closing	Opening. When output is increasing the valve is opening.
15	OUT Value	Output value
16	High OUT Limit	Upper limit output
17	Low OUT Limit	Lower limit output
18	ACT Value	Actuator position
19	Measured Value	The MV object value
19.1	MV Value	AI Error
19.2	MV Unit	Measured unit
20	Setpoint Value	The SP object value
20.1	SP Value	Setpoint value
20.2	SP Unit	Setpoint unit
21	Output Value	The OUT object value
21.1	OUT Value	Output value
21.2	OUT Unit	Output unit

No	Description	Remarks
10	Low Limit 2	Low Alarm limit 2
11	SP Value	Setpoint Value
12	High SP Limit	Upper limit setpoint
13	Low SP Limit	Lower limit setpoint

No	Description	Remarks
22	External OUT Limitation	OUT limited by external limitation
23	Actuator Value	The ACT object value
23.1	ACT Value	Actuator value
23.2	SP Unit	Actuator unit

Appendix B Process Dialog

B.1 General

The Operator Station dialog keys for PIDCONA are described in this appendix. It is reprinted from *AdvaCommand User Interface Reference Manual*.

B.2 Process Dialog for Adaptive Process Controller PIDCONA

The display screen is supplemented with a keyboard for operator communication with the functional unit/object.

The dialog is built up using both fixed and dynamic keys. The functions available with the dynamic keys are presented in a pattern of rectangles on the lower part of the display screen when the object is selected. The functional unit PIDCONA has a MIN dialog (Production level) and a MAX dialog (Maintenance level). The MIN dialog consists of an output dialog with 1 subdialog. The MAX dialog consists of an output dialog with 8 subdialogs.

B.2.1 MIN dialog (No authority key)

Keyboard keys

+ ↑	**Increase**	Increase output value with 0.5% Increase setpoint value with 0.5%
⇧ Shift **+** + ↑		Increase output value with 5.0% Increase setpoint value with 5.0%
↓ −	**Decrease**	Decrease output value with 0.5% Decrease setpoint value with 0.5%
⇧ Shift **+** ↓ −		Decrease output value with 5.0% Decrease setpoint value with 5.0%
Man	**Man**	Set to manual mode
Auto	**Auto**	Set to automatic mode
E1	**E1**	Set to external1 mode
E2	**E2**	Set to external2 mode
✓	**Acknowledge**	Acknowledge alarm
Object	**Object Display**	Select Object Display
📈	**Object Trend**	Select Object Trend
Dialog	**Previous Level**	Select Top Level dynamic keys

Dynamic keys, Top Level

D1 Man Forced	D2 E3	D3	D4	D5
D6 OUT	D7 SP	D8	D9	D10 *Keyboard…*

Man Forced	Set to manual forced mode
E3	Set to external3 mode
OUT	Check/change output value
SP	Check/change setpoint value
Keyboard…	Select Dynamic keys: Keyboard

Dynamic keys, Keyboard

D1 ↑ Increase	D2 Man	D3 Auto	D4 E1	D5 E2
D6 ↓ Decrease	D7 Acknowledge	D8 Object Display	D9 Object Trend	D10

Increase	Increase output value with 0.5%
Shift + Increase	Increase output value with 5.0%
Man	Set to manual mode
Auto	Set to automatic mode
E1	Set to external1 mode
E2	Set to external2 mode
Decrease	Decrease output value with 0.5%
Shift + Decrease	Decrease output value with 5.0%
Acknowledge	Acknowledge alarm
Object Display	Select Object Display
Object Trend	Select Object Trend

B.2.2 MAX dialog (Authority key)

Keyboard keys

+ ↑	**Increase**	Increase output value with 0.5% Increase setpoint value with 0.5%
⇧ Shift + (+ ↑)		Increase output value with 5.0% Increase setpoint value with 5.0%
↓ −	**Decrease**	Decrease output value with 0.5% Decrease setpoint value with 0.5%
⇧ Shift + (↓ −)		Decrease output value with 5.0% Decrease setpoint value with 5.0%
Man	**Man**	Set to manual mode
Auto	**Auto**	Set to automatic mode
E1	**E1**	Set to external1 mode
E2	**E2**	Set to external2 mode
✓	**Acknowledge**	Acknowledge alarm
Object	**Object Display**	Select Object Display
📈	**Object Trend**	Select Object Trend
Dialog	**Previous Level**	Select Top Level dynamic keys
Dialog	**Next Block**	Page between the dialogs on the same level

Dynamic keys, Top Level

D1 Man Forced	D2 E3	D3 *Autotune...*	D4 *Block…*	D5 *Limits…*
D6 OUT	D7 SP	D8 *Adaptation...*	D9 *Parameters…*	D10 *Keyboard…*

Man Forced	Set to manual forced mode
E3	Set to external3 mode
Autotune...	**Select Dynamic keys: Autotune**
Block…	Select Dynamic keys: Block
Limits…	Select Dynamic keys: Limits
OUT	Check/change output value
SP	Check/change setpoint value
Adaptation...	**Select Dynamic keys: Adaptation**
Parameters…	Select Dynamic keys: Parameters
Keyboard…	Select Dynamic keys: Keyboard

Dynamic keys, Keyboard

D1 ↑ Increase	D2 Man	D3 Auto	D4 E1	D5 E2
D6 ↓ Decrease	D7 Acknowledge	D8 Object Display	D9 Object Trend	D10

Increase	Increase output value with 0.5%
Shift + Increase	Increase output value with 5.0%
Man	Set to manual mode
Auto	Set to automatic mode
E1	Set to external1 mode
E2	Set to external2 mode
Decrease	Decrease output value with 0.5%
Shift + Decrease	Decrease output value with 5.0%
Acknowledge	Acknowledge alarm
Object Display	Select Object Display
Object Trend	Select Object Trend

Dynamic keys, Autotune…

D1 Start Tune	D2 Continue Tune	D3 Verify	D4 Accept&Save	D5 Stop&Restore
D6 Response Setup...	D7 Extended Setup...	D8	D9	D10 Retrive Aborted Setting

Start Tune	Start autotuning
Continue Tune	Continue autotuning
Verify	Try to verify setting
Accept&Save	Accept autotuning and save present parameters
Stop&Restore	Stop autotuning and restore present parameters
Response Setup...	Select Dynamic keys: Response Setup
Extended Setup...	Select Dynamic keys: Extended Setup
Retrive Aborted Setting	Retrive aborted setting

Dynamic keys, Adaptation…

D1 Adaptation Enable	D2	D3	D4 Save	D5 Restore
D6 Adaptation Off	D7	D8	D9	D10

Adaptation Enable	Enable adaptation
Save	Saving present parameters
Restore	Restoring present parameters
Adaptation Off	Disable adaptation

Dynamic keys, Block…

D1 Block Alarm1	D2 Block Printout1	D3	D4 Block Alarm2	D5 Block Printout2
D6 Deblock Alarm1	D7 Deblock Printout	D8	D9 Deblock Alarm2	D10 Deblock Printout2

Block Alarm1	Block alarm1 indication
Block Printout1	Block event1 printout
Block Alarm2	Block alarm2 indication
Block Printout2	Block event2 printout
Deblock Alarm1	Deblock alarm1 indication
Deblock Printout1	Deblock event1 printout
Deblock Alarm2	Deblock alarm2 indication
Deblock Printout2	Deblock event2 printout

Dynamic keys, Limits…

D1 H1 80.00	D2 H2 90.00	D3 SP H 50.00	D4 DEV H1 2.20	D5 OUT H 100.00
D6 L1 20.00	D7 L2 10.00	D8 SP L 10.00	D9 DEV L1 -1.90	D10 OUT L 0.00

H1	Dialog for changing measured value warning limit H1
H2	Dialog for changing measured value alarm limit H2
SP H	Dialog for changing upper warning limit setpoint
DEV H1	Dialog for changing upper warning limit for deviation
OUT H	Dialog for changing upper warning limit output
L1	Dialog for changing measured value warning limit L1
L2	Dialog for changing measured value alarm limit L2
SP L	Dialog for changing lower warning limit setpoint
DEV L1	Dialog for changing lower warning limit for deviation
OUT L	Dialog for changing lower warning limit output

Dynamic keys, Parameters…

D1	D2	D3 Gain 1.32	D4 TI 6.01	D5 TD 4.01
D6	D7	D8 BETA 0.346	D9 TF 2.45	D10 DZ 0.98

Gain	Dialog for changing gain
TI	Dialog for changing the time constant for integration
TD	Dialog for changing the time constant for derivation
BETA	Dialog for changing the time constant
TF	Dialog for changing the filter time
DZ	Dialog for changing deadzone

Dynamic keys, Response Setup…

D1 Extra Damped	D2 Damped	D3 Normal	D4 Fast	D5 Extra Fast
D6	D7	D8	D9	D10

Extra Damped	Set response to extra damped
Damped	Set response to damped
Normal	Set response to normal
Fast	Set response to fast
Extra Fast	Set response to extra fast

Dynamic keys, Extended Setup…

D1 Excitation Level OUT 6.00	D2 Excitation Level SP 4.50	D3	D4 Frequency Limit 0.398	D5 Frequency Unlimited
D6	D7	D8	D9	D10

Excitation Level OUT	Dialog for changing Excitation Level Output
Excitation Level SP	Dialog for changing Excitation Level Setpoint
Frequency Limit	Dialog for changing Frequency Limit
Frequency Unlimit	Dialog for changing to Frequency Unlimit

Appendix C PC Element

C.1 General

This description of the PIDCONA PC element is reprinted from *PC Elements Advant Controller 400 Series Reference Manual*.

Table C-1. Typical PC element execution time with different extensions included

AC 450, Execution Time, typical (µs)	Extensions included
400	None
420	C1...C3
450	C1...C4
550	C1...C5
750	C1...C6
520	C4...C5

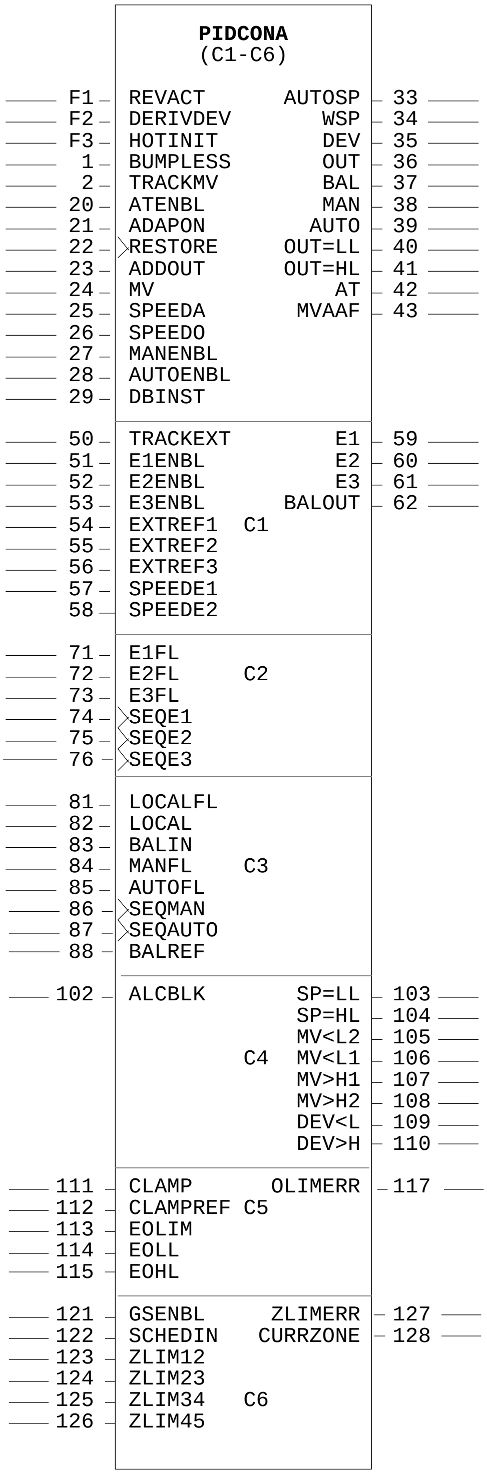
Table C-2. Memory requirement

PC statement (byte)	Local data area (byte)
66+26×C1+12×C2+16×C3+18×C4 +12×C5+16×C6	14 (for QC02-LIB21*7.0/0: 2162)

Process Controller with AutotuningPIDCONA

Summary

The PC element PIDCONA forms part of the functional unit PIDCONA (**PID CON**troller with built in **A**utotuning and adaptive control). The unit is a flexible and powerful process controller with standardized operator functions. It is used in traditional instrumentation when a VDU based operator station is specified.



PC Element PIDCONA

Call PIDCONA (C1, C2, C3, C4, C5, C6)

The element consists of one basic part that gives a regulator function in its basic form. This part may be extended with optional functions which are selected using call parameters C1–C6, see table below.

Parameter	Significance	Permissible values
C1	Ordering of operation mode E1, E2 or E3.	0 .. 1
C2	Ordering of operation mode E1, E2 or E3 via PC connection	0 .. 1 (1 requires C1=1)
C3	Ordering of operation mode BAL, MAN or AUTO via PC connection.	0 .. 1
C4	Supervision against alarm limits	0 .. 1
C5	Output signal control	0 .. 1
C6	Gain scheduling	0 .. 1

Terminal Description

No	Name	Type	Description
F1	REVACT	II	**REV**erse **ACT**ion. Function parameter for control action. Reverse action (REVACT =1) means that the output signal (OUT) decreases with increasing measured value (MV) (the normal case). Direct action (REVACT=0) means that the output signal increases with increasing measured value (used when a change of the manipulated variable (OUT) leads to a final change of the controlled variable (MV) in the opposite direction).
F2	DERIVDEV	II	**DERIV**ation of the **DEV**iation. Function parameter for selection of whether D action is performed on the measured value (MV) or on the deviation (DEV). DERIVDEV=0 is normally recommended. It yields D action on the measured value (MV). DERIVDEV=1 yields D action on the deviation (DEV).
F3	HOTINIT	II	**HOT INIT**. Function parameter determining the state after a warm start. HOTINIT=0 is chosen, unless special actions are required. HOTINIT=0 gives unchanged mode and unchanged output. HOTINIT=1 gives mode ”MAN” and unchanged output. HOTINIT=2 gives mode ”MAN” and OUT=0. If 0 is outside the limits, OUT is set to the closest limit. By unchanged is meant no change in relation to the state in effect before the event that caused the warm start (the conditions are described in the manual Functional Units, Part1).
1	BUMPLESS	IB	**BUMPLESS** transfer. Input for selection of bumpless transfer from mode BAL, MAN or E3 to mode AUTO, E1, or E2. With BUMPLESS=1 there will be no ”P step” on transition, and the working setpoint WSP is ramped from the value giving no output change to the value for the new mode, using that mode's setpoint rate limit. With BUMPLESS=0 the new setpoint will be used directly, without ramping to it, and there will be a ”P step” if there is an initial deviation.
2	TRACKMV	IB	**TRACK** the **M**easured **V**alue. Input for selection of tracking or no tracking for the operator setpoint AUTOSP in the modes BAL, MAN, and E3. With TRACKMV=0, there is no tracking, and AUTOSP is available for operator changes in these modes. With TRACKMV=1, AUTOSP tracks the measured value in these modes, thereby making the transition to mode AUTO free from output change.
20	ATENBL	IB	**A**uto**T**uning **EN**a**BL**e. Input for allowing autotuning. If ATENBL=1, autotuning is allowed. If ATENBL=0, autotuning is interrupted and can not be started.
21	ADAPON	IB	**ADAP**tive control **ON**. Input for requesting the control to be adaptive in modes AUTO, E1, and E2.

Terminal Description

No	Name	Type	Description
22	RESTORE	IB	**RESTORE** saved controller settings. Dynamic input. A positive transition causes the saved tuning (of the current zone, if C6=1) to replace the present one.
23	ADDOUT	IR	**ADD**ed **OUT**put value. Input that is added to the control algorithm output. Used for bias, when there is no integral action, or for feedforward.
24	MV	IR	**M**easured **V**alue. Input for the measured value (the feedback signal representing the controlled variable).
25	SPEEDA	IR	Input for the largest change rate for the setpoint in AUTO mode (engineering unit/second).
26	SPEEDO	IR	Input for the largest change rate for the output (OUT) (engineering unit/second). Applies to all modes except BAL.
27	MANENBL	IB	**MAN**ual **EN**a**BL**e. Input for enabling ordering of operation mode MAN by the operator. 0=MAN may not be ordered by the operator. 1=MAN may be ordered by the operator.
28	AUTOENBL	IB	**AUTO EN**a**BL**e. Input for enabling ordering of operation mode AUTO by the operator. 0=AUTO may not be ordered by the operator. 1=AUTO may be ordered by the operator.
29	DBINST	II	**D**ata **B**ase **INST**ance. An operational parameter or a constant, defining the instance number of the data base (to be found in its header). Alternatively, a connection to the data base element.
33	AUTOSP	OR	**AUTO S**et **P**oint. Output for setpoint in operation mode AUTO.
34	WSP	OR	**W**orking **S**et**P**oint. Output for working setpoint.
35	DEV	OR	**DEV**iation. Output for control deviation.
36	OUT	OR	**OUT**put. Output signal of the controller, the manipulated variable.
37	BAL	OB	**BAL**ance. Output for operation mode indication. This output is set to 1 with operation mode BAL.
38	MAN	OB	**MAN**ual. Output for operation mode indication. This output is set to 1 with operation mode MAN.
39	AUTO	OB	**AUTO**. Output for operation mode indication. This output is set to 1 with operation mode AUTO.
40	OUT=LL	OB	**OUT**put**=L**ow **L**imit. Output for indication of output signal limitation. It is set to 1 when the output signal equals lower limit value.
41	OUT=HL	OB	**OUT**put**=H**igh **L**imit. Output for indication of output signal limitation. It is set to 1 when the output signal equals the upper limit value.
42	AT	OB	**A**uto**T**uning. Output indicating 1 during autotuning, otherwise 0.
43	MVAAF	OR	**M**easured **V**alue, **A**nti-**A**lias **F**iltered. Output for the measured value after anti-alias filtering and sampling, with the rate presently used in the control algorithm.
50(1)	TRACKEXT	II	**TRACK** the **EXT**ernal reference. Input for selection of tracking or no tracking for the operator set point AUTOSP in the modes E1 and E2. With TRACKTEXT=0, there is no tracking, and AUTOSP is available for operator changes in these modes. With TRACKTEXT=1, AUTOSP tracks the working set point WSP. This gives change-free transitions from E1 and E2 to AUTO. With TRACKTEXT=2, AUTOSP tracks EXTREF1 in mode E1, and EXTREF2 in mode E2. Thereby a set point ramp started in E1 or E2 will continue to to its final value (with rate SPEEDA) after transition to AUTO.

Terminal Description

No	Name	Type	Description
51(1)	E1ENBL	IB	**E1 EN**a**BL**e. Input for enabling ordering of operation mode E1 by the operator. 0=E1 may not be ordered by the operator. 1=E1 may be ordered by the operator.
52(1)	E2ENBL	IB	**E2 EN**a**BL**e. Input for enabling ordering of operation mode E2 by the operator. 0=E2 may not be ordered by the operator. 1=E2 may be ordered by the operator.
53(1)	E3ENBL	IB	**E3 EN**a**BL**e. Input for enabling ordering of operation mode E3 by the operator. 0=E3 may not be ordered by the operator. 1=E3 may be ordered by the operator.
54(1)	EXTREF1	IR	**EXT**ernal **REF**erence **1**. Input for external reference (setpoint) in operation mode E1.
55(1)	EXTREF2	IR	**EXT**ernal **REF**erence **2**. Input for external reference (setpoint) in operation mode E2.
56(1)	EXTREF3	IR	**EXT**ernal **REF**erence **3**. Input for external reference for the output OUT, in operation mode E3.
57(1)	SPEEDE1	IR	Input for the maximum change rate for the setpoint in mode E1 (engineering unit/second).
58(1)	SPEEDE2	IR	Input for the maximum change rate for the setpoint in mode E2 (engineering unit/second).
59(1)	E1	OB	Output for operation mode indication. This output is set to 1 with operation mode **E1**.
60(1)	E2	OB	Output for operation mode indication. This output is set to 1 with operation mode **E2**.
61(1)	E3	OB	Output for operation mode indication. This output is set to 1 with operation mode **E3**.
62(1)	BALOUT	OB	**BAL**ance **OUT**. Output for operation mode indication. This output is set to 1 with operation modes BAL, MAN and AUTO.
71(2)	E1FL	IB	**E1 FL**ash. Input for ordering of operation mode E1. It also gives an event report.
72(2)	E2FL	IB	**E2 FL**ash. Input for ordering of operation mode E2. It also gives an event report.
73(2)	E3FL	IB	**E3 FL**ash. Input for ordering of operation mode E3. It also gives an event report.
74(2)	SEQE1	IB	**SEQ**uential **E1**. Dynamic input for ordering of operation mode E1.
75(2)	SEQE2	IB	**SEQ**uential **E2**. Dynamic input for ordering of operation mode E2.
76(2)	SEQE3	IB	**SEQ**uential **E3**. Dynamic input for ordering of operation mode E3.
81(3)	LOCALFL	IB	**LOCAL FL**ash. Input for ordering of operation mode BAL. It gives an event report and VDU indication BAL LOCAL.
82(3)	LOCAL	IB	**LOCAL**. Input for ordering of operation mode BAL. It gives VDU indication BAL LOCAL.
83(3)	BALIN	IB	**BAL**ance **IN**put. Input for ordering of operation mode BAL.
84(3)	MANFL	IB	**MAN**ual **FL**ash. Input for ordering of operation mode MAN. It gives an event report.
85(3)	AUTOFL	IB	**AUTO FL**ash. Input for ordering of operation mode AUTO. It gives an event report.
86(3)	SEQMAN	IB	**SEQ**uential **MAN**ual. Dynamic input for ordering of operation mode MAN.
87(3)	SEQAUTO	IB	**SEQ**uential **AUTO**. Dynamic input for ordering of operation mode AUTO.
88(3)	BALREF	IR	**BAL**ance **REF**erence. Input for reference value for the manipulated variable (the output signal, OUT) with operation mode BAL.
102(4)	ALCBLK	IB	**AL**arm **C**ontrol **BL**oc**K**. Input which, when set to 1, blocks limit value alarms for MV and DEV.

Terminal Description

No	Name	Type	Description
103(4)	SP=LL	OB	**S**et**P**oint**=L**ow **L**imit. Output set to 1 when the setpoint equals the lower limit value.
104(4)	SP=HL	OB	**S**et **P**oint**=H**igh **L**imit. Output set to 1 when the set point equals the upper limit value.
105(4)	MV<L2	OB	**M**easured **V**alue<**L**ow **2**. Output set to 1 when the measured value (MV) falls below alarm limit L2.
106(4)	MV<L1	OB	**M**easured **V**alue<**L**ow **1**. Output set to 1 when the measured value (MV) falls below alarm limit L1.
107(4)	MV>H1	OB	**M**easured **V**alue>**H**igh **1**. Output set to 1 when the measured value exceeds alarm limit H1.
108(4)	MV>H2	OB	**M**easured **V**alue>**H**igh **2**. Output set to 1 when the measured value exceeds alarm limit H2.
109(4)	DEV<L	OB	**DEV**iation<**L**ow. Output set to 1 when the deviation (DEV) falls below the low alarm limit (L).
110(4)	DEV>H	OB	**DEV**iation>**H**igh. Output set to 1 when the deviation (DEV) exceeds the high alarm limit (H).
111(5)	CLAMP	IB	**CLAMP** the output (OUT) to its clamp reference value. Input for ordering of operation mode MAN with clamping. The mode and the output can not be changed by the operator in this mode.
112(5)	CLAMPREF	IR	**CLAMP REF**erence. Clamp value with operation mode MAN with clamping.
113(5)	EOLIM	IB	**E**xternal **O**utput **LIM**it. Change signal for output signal limits. 0=from the operator's keyboard. 1=from the inputs EOLL and EOHL.
114(5)	EOLL	IR	**E**xternal **O**utput **L**ow **L**imit. Input for lower limit for the output signal.
115(5)	EOHL	IR	**E**xternal **O**utput **H**igh **L**imit. Input for upper limit for the output signal.
117(5)	OLIMERR	OB	**O**utput **LIM**it **ERR**or. Output which, when set to 1, indicates an error in the alarm limits, i.e. EOLL>EOHL.
121(6)	GSENBL	IB	**G**ain **S**cheduling **EN**a**BL**e. Input to select gain scheduling active (GSENBL=1), or turned off (GSENBL=0). When active, saved settings are restored on changes of current zone. When turned off, zone 1 is considered current independent of SCHEDIN. The mere change of GSENBL does not cause any settings to be restored.
122(6)	SCHEDIN	IR	**SCHED**uling **IN**put. Input for the gain scheduling variable. Its value, compared to the zone limits, determines the current zone.
123(6)	ZLIM12	IR	**Z**one **LIM**it between zones **1** and **2**. Input determining this gain scheduling zone limit.
124(6)	ZLIM23	IR	**Z**one **LIM**it between zones **2** and **3**. Input determining this gain scheduling zone limit.
125(6)	ZLIM34	IR	**Z**one **LIM**it between zones **3** and **4**. Input determining this gain scheduling zone limit.
126(6)	ZLIM45	IR	**Z**one **LIM**it between zones **4** and **5**. Input determining this gain scheduling zone limit.
127(6)	ZLIMERR	OB	**Z**one **LIM**it **ERR**or. Output which, when set to 1, indicates an error in the relations between the zone limits. It is required that the limits are ascending, i.e. ZLIM12≤ZLIM23≤ZLIM34≤ZLIM45.
128(6)	CURRZONE	OI	CURRent ZONE. Output indicating the current zone of gain scheduling. It equals 1, if gain scheduling is not enabled.

(1) Available only if C1=1.
(2) Available only if C2=1.
(3) Available only if C3=1.
(4) Available only if C4=1.
(5) Available only if C5=1.
(6) Available only if C6=1.

Appendix D Data Base Elements

D.1 General

This description of the data base elements for PIDCONA is reprinted from *Data Base Elements Advant Controller 400 Series Reference Manual*.

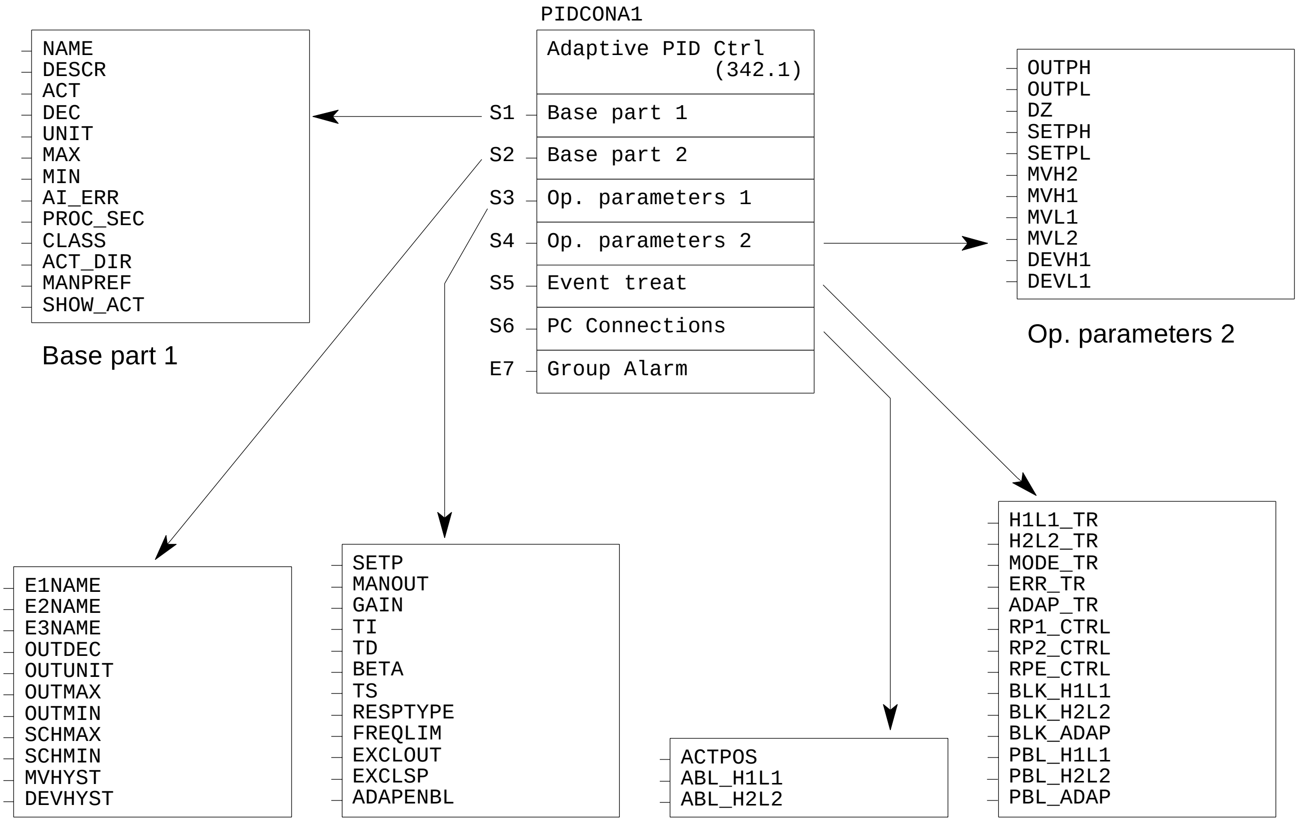
Adaptive PID Ctrl PIDCONA

Summary

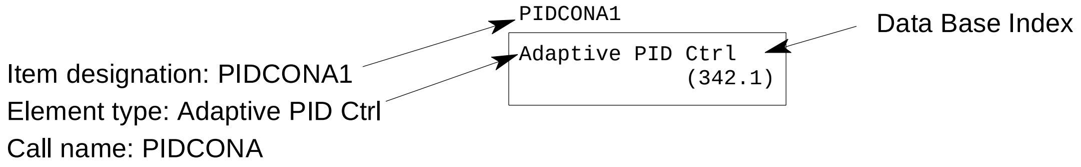
The Adaptive PID control data base element is part of the functional unit PIDCONA. It is used to set up the functions and the presentation for the operator station, such as names, ranges, units, and alarm handling. Initial values can be specified for some of the parameters that are normally entered via operator station dialog. The element also provides an interface between the operator station functions and the PC program during normal operation.

For information about the extension part Group Alarm, see data base element GRPMEMB.

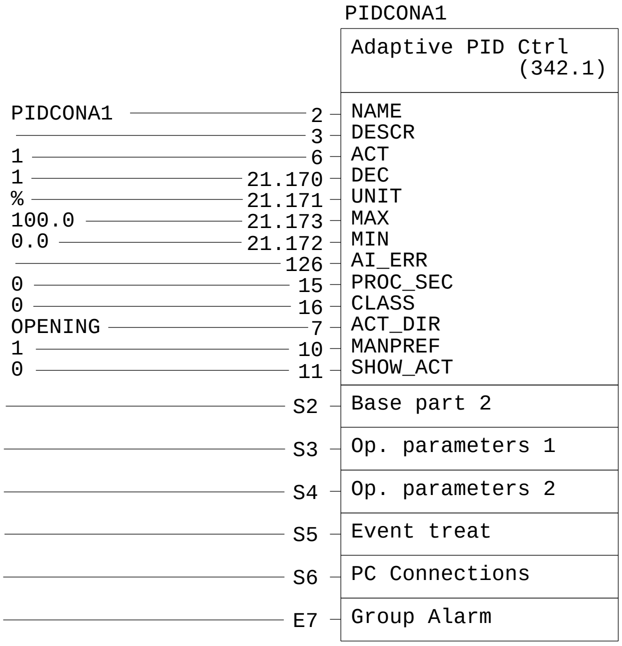
Overview



Head



Base part 1



Terminal Description, Base part 1

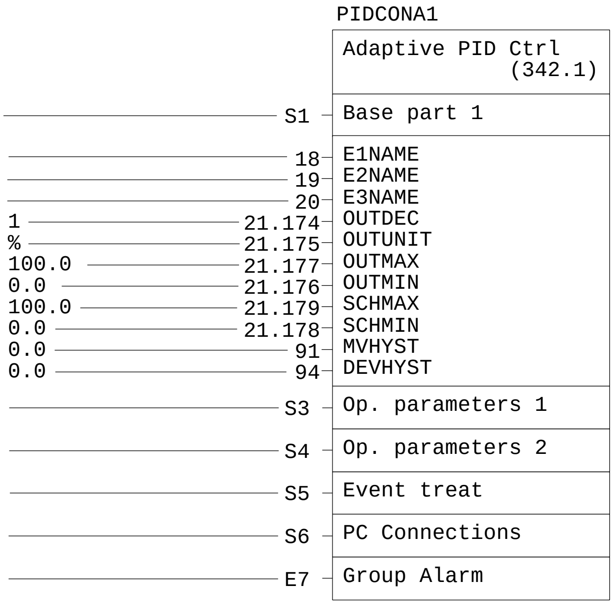
Termi-nal No	Terminal Name	Value entered by	Default value	PC con-nection data type	Description	Remarks
2	NAME	user	PIDCONAx	–	Unique **NAME** of process controller (e.g. TICA107) where x=instance number.	–
3	DESCR	user	–	–	This **DESCR**iption of the Controller will be shown in operator station.	–
6	ACT	user	1	B(r/w)	1=the operator communication is **ACT**ive.	–

Terminal Description, Base part 1 (Continued)

Terminal No	**Terminal Name**	**Value entered by**	**Default value**	**PC connection data type**	**Description**	**Remarks**
21.170	DEC	user	1	–	The number of **DEC**imals for measured value and its alarm limits, for set points and their limits, and for range min and range max.	0..6 decimals.
21.171	UNIT	user	%	A6(r)	The **UNIT** of the measured value and set point in engineering unit.	–
21.173	MAX	user	100.0	R(r/w)	The **MAX**imum value of the measuring range.	(1)
21.172	MIN	user	0.0	R(r/w)	The **MIN**imum value of the measuring range.	(1)
126	AI_ERR	user	–	–	**AI ERR**or. Reference to the Analog Input element which is used for the measured value.	Connects error flag in the analog input to PIDCONA. E.g. FT 107. An initiation must be made in theAdvant Controller 400 Series system before the change is accomplished.
15	PROC_SEC	user	0	I(r/w)	**PROC**ess **SEC**tion is used for sectioning the alarm handling. Section numbers 1-16.	0=no sectioning. The signal may be presented in lists –1=the signal cannot be selected and cannot be presented in lists.
16	CLASS	user	0	I(r/w)	**CLASS** subdivides process section. Used by status list function in Operator Station	–
7	ACT_DIR	user	OPENING	B(r/w)	**ACT**uator **DIR**ection. Used for presentation in the object and group displays. OPENING=actuator open at 100% output. CLOSING=actuator closed at 100% output.	–
10	MANPREF	user	1	B(r)	Presently unused. Reserved for future use.	–
11	SHOW_ACT	user	0	B(r/w)	**SHOW ACT**uator position (bar graph and value) in displays.	–

(1) MIN < MAX is required.

Base part 2



Terminal Description, Base part 2

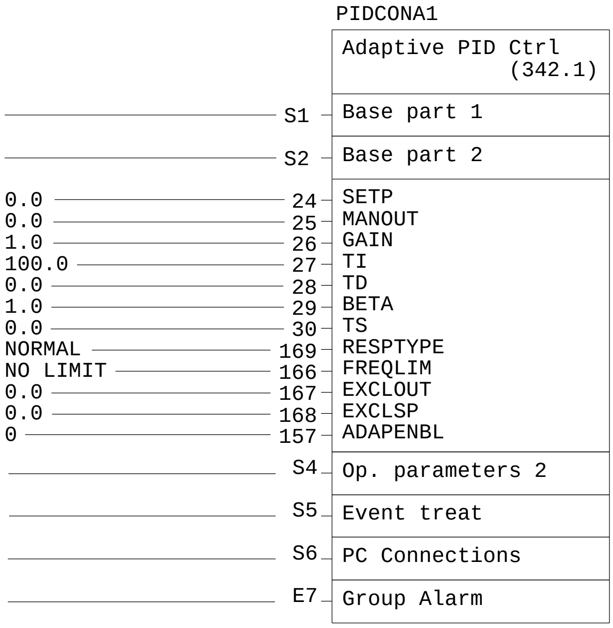
Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
18	E1NAME	user	–	–	**NAME** of mode **E1** shown in the object display.	Max. 20 characters.
19	E2NAME	user	–	–	**NAME** of mode **E2** shown in the object display.	Max. 20 characters.
20	E3NAME	user	–	–	**NAME** of mode **E3** shown in the object display.	Max. 20 characters.
21.174	OUTDEC	user	1	–	The number of **DEC**imals for **OUT**put and its alarm limits and range min and range max.	–
21.175	OUTUNIT	user	%	A6(r)	The **UNIT** of the **OUT**put value in engineering unit.	–
21.177	OUTMAX	user	100.0	R(r/w)	The **MAX**imum value of the **OUT**put range.	(1)
21.176	OUTMIN	user	0.0	R(r/w)	The **MIN**imum value of the **OUT**put range.	(1)
21.179	SCHMAX	user	100.0	R(r/w)	The **MAX**imum value of the **SCH**ed in range.	(2)

Terminal Description, Base part 2 (Continued)

Terminal No	**Terminal Name**	**Value entered by**	**Default value**	**PC connection data type**	**Description**	**Remarks**
21.178	SCHMIN	user	0.0	R(r/w)	The **MIN**imum value of the **SCH**ed in range.	(2)
91	MVHYST	user	0.0	R(r/w)	**HYST**eresis for alarm limits of **M**easured **V**alue.	≥0
94	DEVHYST	user	0.0	R(r/w)	**HYST**eresis for alarm limits for **DEV**iation.	≥0

(1) OUTMIN < OUTMAX is required.
(2) SCHMIN < SCHMAX is required.

Op. parameters 1



Terminal Description, Op. parameters 1

Terminal No	**Terminal Name**	**Value entered by**	**Default value**	**PC connection data type**	**Description**	**Remarks**
24	SETP	dialog	0.0	R(r/w)	Auto **SETP**oint.	–
25	MANOUT	dialog	0.0	R(r/w)	**MAN**ual **OUT**put.	–

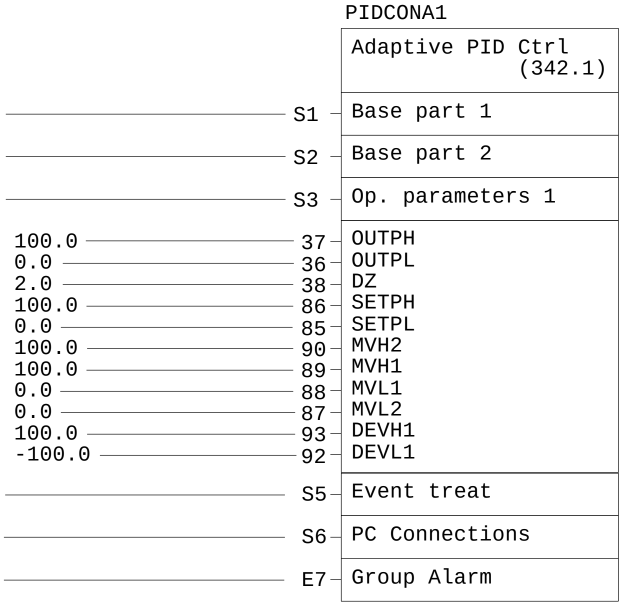
Terminal Description, Op. parameters 1 (Continued)

Terminal No	**Terminal Name**	**Value entered by**	**Default value**	**PC connection data type**	**Description**	**Remarks**
26	GAIN	system/ dialog	1.0	R(r/w)	**GAIN** value.	–
27	TI	system/ dialog	100.0	R(r/w)	**I**ntegration **T**ime in seconds.	≥0
28	TD	system/ dialog	0.0	R(r/w)	**D**erivation **T**ime in seconds.	≥0
29	BETA	system/ dialog	1.0	R(r/w)	Setpoint factor (**BETA**) in P part.	≥0
30	TS	system/ dialog	0.0	–	**S**ampling **T**ime (period) in seconds.	≥0
169	RESPTYPE	dialog	NORMAL	–	**RESP**onse **TYPE**. EXTRA FAST, FAST, NORMAL, DAMPED, EXTRA DAMPED	–
166	FREQLIM	dialog	NO LIMIT	–	Maximum speed of dominant poles. "No limit" is shown for values 10^6 or greater, other values numerically. "No limit" entered results in 10^6 being stored.	>0
167	EXCLOUT	dialog	0.0	–	**EXC**itation **L**evel for the **OUT**put. (1)	≥0
168	EXCLSP	dialog	0.0	–	**EXC**itation **L**evel for the **S**et**P**oint. (2)	≥0
157	ADAPENBL	dialog	0	–	**ADAP**tation **EN**a**BL**ed flag.	–

(1) 0 is interpreted as the default, which is 3% of output range OUTMAX-OUTMIN. Positive values are the level in engineering units.

(2) 0 is interpreted as the default, which is 3% of measured value range MAX-MIN. Positive values are the level in engineering units.

Op. parameters 2



Terminal Description, Op. parameters 2

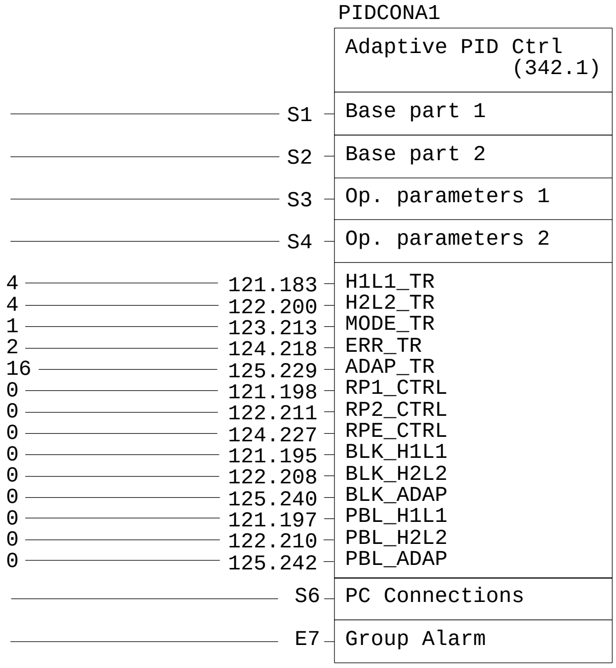
Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
37	OUTPH	dialog	100.0	R(r/w)	**OUTP**ut **H**igh limit.	(1)
36	OUTPL	dialog	0.0	R(r/w)	**OUTP**ut **L**ow limit.	(1)
38	DZ	dialog	0.0	R(r/w)	**D**ead **Z**one (%). When a three point controller CON-PU1 is used, it can get its deadzone from this terminal. The value is also an adaption deadzone for the PIDCONA output. No adaption when the output varies less.	≥0
86	SETPH	dialog	100.0	R(r/w)	**H**igh limit of **SETP**oint.	(2)
85	SETPL	dialog	0.0	R(r/w)	**L**ow limit of **SETP**oint.	(2)
90	MVH2	dialog	100.0	R(r/w)	Alarm limit **H2** of **M**easured **V**alue.	(3)
89	MVH1	dialog	100.0	R(r/w)	Alarm limit **H1** of **M**easured **V**alue.	(3)
88	MVL1	dialog	0.0	R(r/w)	Alarm limit **L1** of **M**easured **V**alue.	(3)
87	MVL2	dialog	0.0	R(r/w)	Alarm limit **L2** of **M**easured **V**alue.	(3)

Terminal Description, Op. parameters 2 (Continued)

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
93	DEVH1	dialog	100.0	R(r/w)	**H**igh alarm limit **1** for control **DEV**iation.	(4)
92	DEVL1	dialog	–100.0	R(r/w)	**L**ow alarm limit **1** for control **DEV**iation.	(4)

(1) OUTPL ≤ OUTPH is required.
(2) SETPL ≤ SETPH is required.
(3) MVL2 ≤ MVL1 ≤ MVH1 ≤ MVH2 is required.
(4) DEVL1 ≤ DEVH1 is required.

Event treat



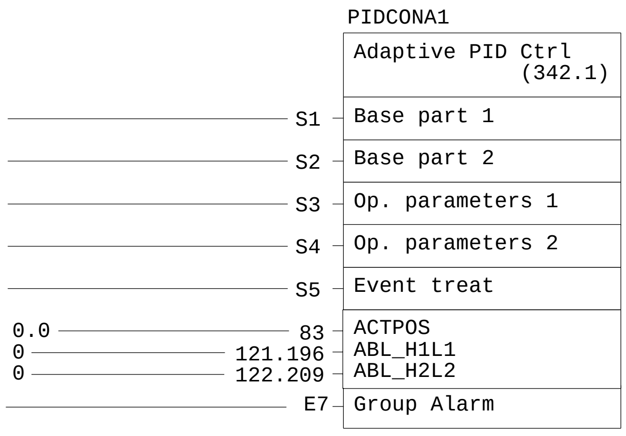
Terminal Description, Event treat

Termi-nal No	**Terminal Name**	**Value entered by**	**Default value**	**PC con-nection data type**	**Description**	**Remarks**
121.183	H1L1_TR	user	4	–	Alarm **H1** and **L1 TR**eatment. Instance number of the data base element Event Treat located in the operator station and defining event treatment of the alarms H1, L1, and control deviation.	–
122.200	H2L2_TR	user	4	–	Alarm **H2** and **L2 TR**eatment. Instance number of the data base element Event Treat located in the operator station and defining event treatment of the alarms H2 and L2.	–
123.213	MODE_TR	user	1	–	**MODE TR**eatment. Instance number of the data base element Event Treat located in the operator station and defining event treatment of the control mode changes done by the PC program.	–
124.218	ERR_TR	user	2	–	**ERR**or **TR**eatment. Instance number of the data base element Event Treat located in the operator station and defining event treatment of signal error in the connected analog input channel (AI–ERR).	–
125.229	ADAP_TR	user	16	–	**ADAP**tion **TR**eatment. Instance number of the data base element Event Treat located in the operator station and defining event treatment of adaption errors.	–
121.198	RP1_CTRL	user	0	–	**R**e**P**eat fail **C**on**TR**o**L** for alarms H**1** and L1. Blocking of repeating H1 and L1 alarms.	–
122.211	RP2_CTRL	user	0	–	**R**e**P**eat fail **C**on**TR**o**L** for alarms H**2** and L2. Blocking of repeating H2 and L2 alarms.	–
124.227	RPE_CTRL	user	0	–	**R**e**P**eat fail **C**on**TR**o**L** for blocking **E**rror. Blocking of repeating signal error in the analog input connected to the controller	–
121.195	BLK_H1L1	dialog	0	–	**BL**oc**K** the alarms **H1**, **L1** and control deviation.	–
122.208	BLK_H2L2	dialog	0	–	**BL**oc**K** the alarms **H2** and **L2**.	–
125.240	BLK_ADAP	user	0	–	**BL**oc**K** the **ADAP**tion alarms.	–

Terminal Description, Event treat (Continued)

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
121.197	PBL_H1L1	dialog	0	–	**BL**ock the **P**rinting of **H1**, **L1** and control deviation alarms.	–
122.210	PBL_H2L2	dialog	0	–	**BL**ock the **P**rinting of **H2** and **L2** alarms.	–
125.242	PBL_ADAP	user	0	–	**BL**ock the **P**rinting of **ADAP**tion alarms.	–

PC Connections



Terminal Description, PC Connections

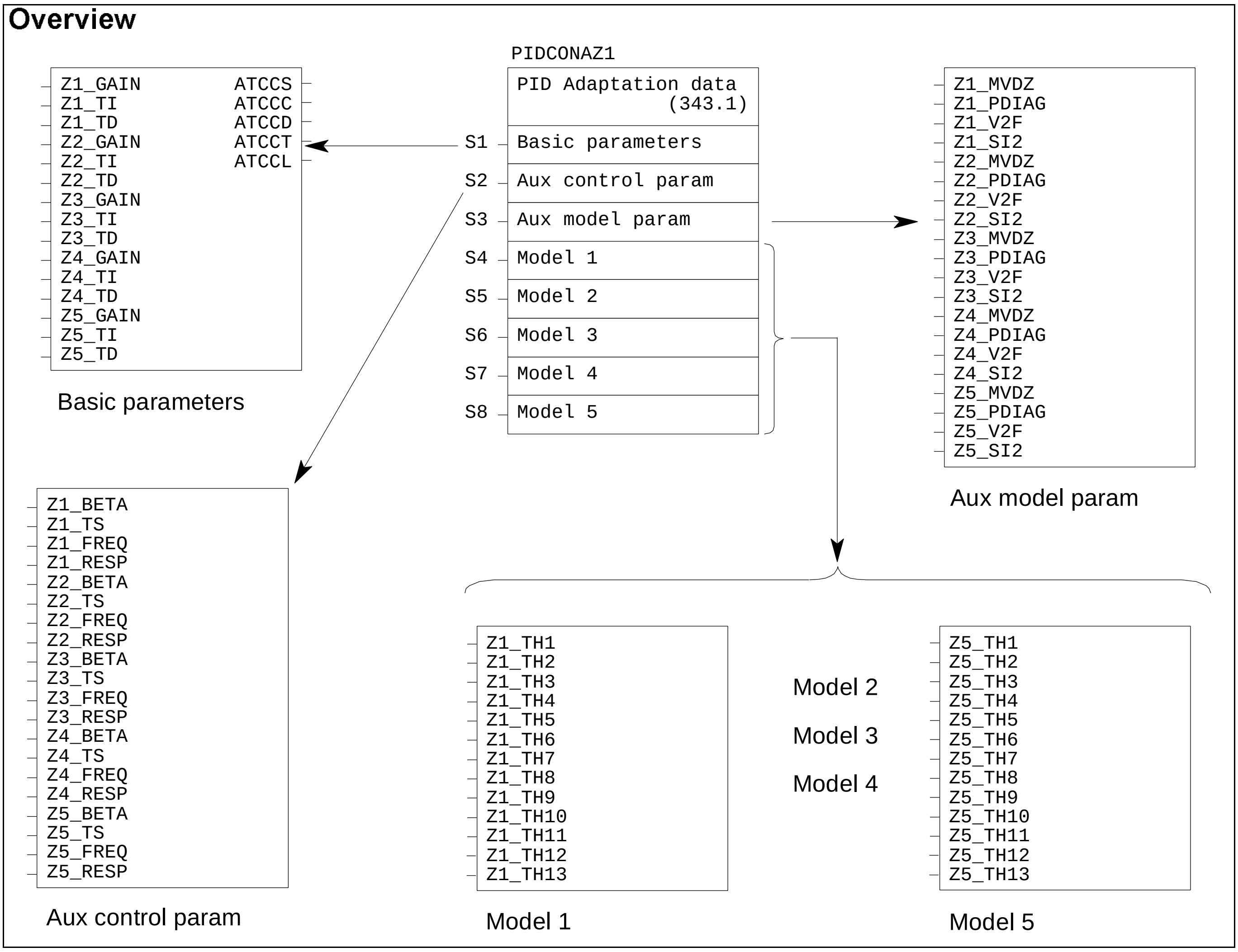
Termi-nal No	**Terminal Name**	**Value entered by**	**Default value**	**PC con-nection data type**	**Description**	**Remarks**
83	ACTPOS	PC	0.0	R(r/w)	To be connected to the AI channel measuring the **ACT**uator **POS**ition (Transferred by a PC element).	The value will be shown in the object display if SHOW_ACT=1.
121.196	ABL_H1L1	PC	0	B(r/w)	**A**uto **BL**ock alarms **H1** and **L1**.	–
122.209	ABL_H2L2	PC	0	B(r/w)	**A**uto **BL**ock alarms **H2** and **L2**.	–

PID Adaption data **PIDCONAZ**

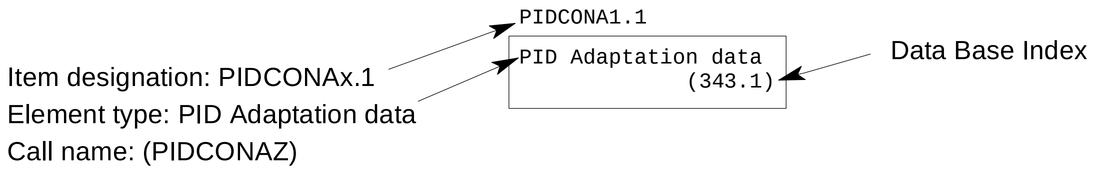
Summary

There are no values to be entered by the user in this data base element. It is subordinated to the Adaptive PID Ctrl data base element.

Saved parameters for the five different gain scheduling zones are available in the element PID Adaptation data, in order to be retained during source code dumping and loading. The well known controller parameters are gathered in the part Basic parameters together with terminals holding completion codes for the latest autotuning. For example, if an autotuning was completed successfully without any operator intervention, the ATCCS terminal of the data base element will show the string SVV. The other completion code terminals give further information about autotuning.



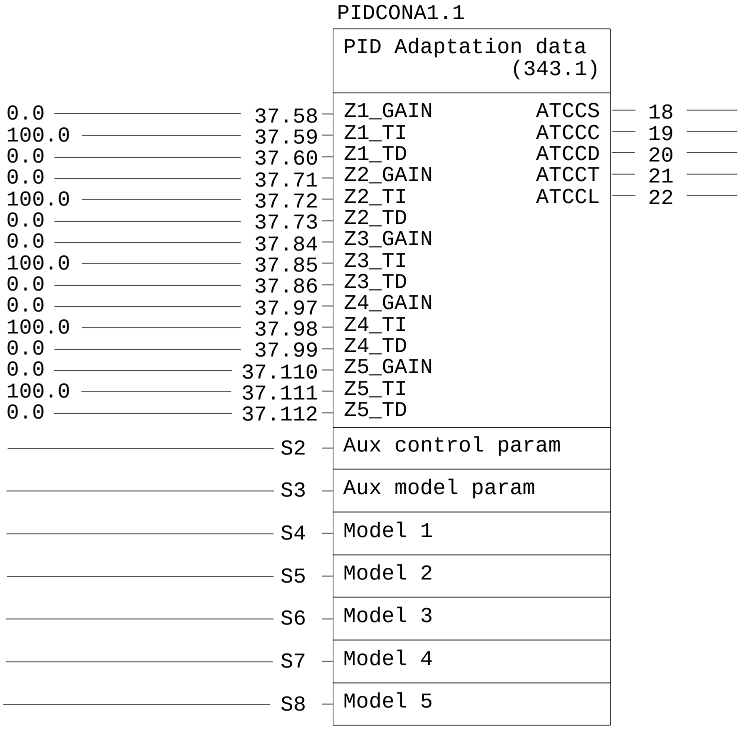
Head



Item designation: PIDCONAx.1
Element type: PID Adaptation data
Call name: (PIDCONAZ)

Data Base Index

Basic parameters



Terminal Description, Basic parameters

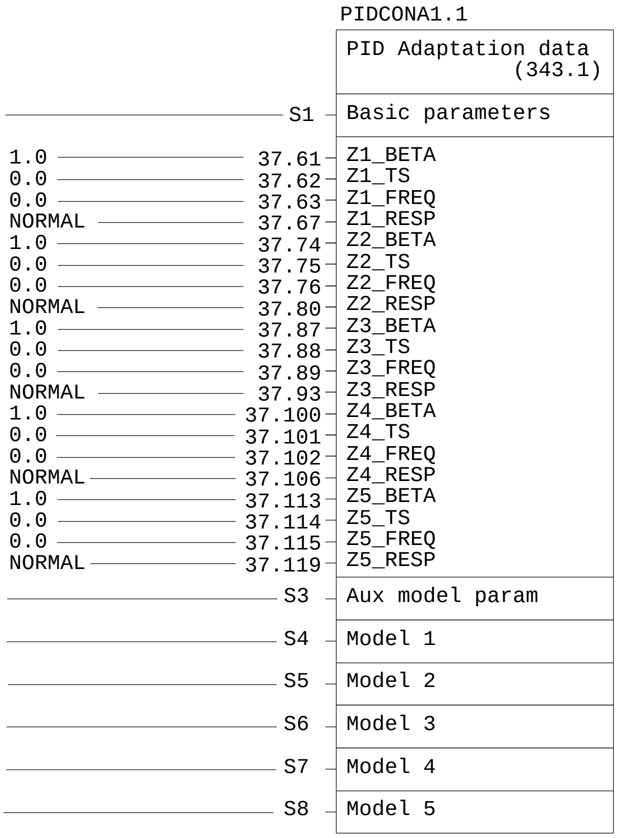
Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
37.58	Z1_GAIN	system	0.0	–	**GAIN** saved in **Z**one **1**.	(1)
37.59	Z1_TI	system	100.0	–	**TI** saved in **Z**one **1**.	≥0
37.60	Z1_TD	system	0.0	–	**TD** saved in **Z**one **1**.	≥0
37.71	Z2_GAIN	system	0.0	–	**GAIN** saved in **Z**one **2**.	(1)

Terminal Description, Basic parameters (Continued)

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
37.72	Z2_TI	system	100.0	–	**TI** saved in **Z**one **2**.	≥0
37.73	Z2_TD	system	0.0	–	**TD** saved in **Z**one **2**.	≥0
37.84	Z3_GAIN	system	0.0	–	**GAIN** saved in **Z**one **3**.	(1)
37.85	Z3_TI	system	100.0	–	**TI** saved in **Z**one **3**.	≥0
37.86	Z3_TD	system	0.0	–	**TD** saved in **Z**one **3**.	≥0
37.97	Z4_GAIN	system	0.0	–	**GAIN** saved in **Z**one **4**.	(1)
37.98	Z4_TI	system	100.0	–	**TI** saved in **Z**one **4**.	≥0
37.99	Z4_TD	system	0.0	–	**TD** saved in **Z**one **4**.	≥0
37.110	Z5_GAIN	system	0.0	–	**GAIN** saved in **Z**one **5**.	(1)
37.111	Z5_TI	system	100.0	–	**TI** saved in **Z**one **5**.	≥0
37.112	Z5_TD	system	0.0	–	**TD** saved in **Z**one **5**.	≥0
18	ATCCS	system	–	–	**C**ompletion **C**ode of latest **A**uto**T**uning. **S**tatus. Three character code (using ASCII). 1st - How AT was started S=Start tune C=Continue tune 2nd - What ended AT M=Man forced or stop & restore A=Accept & save V=Verification successful F=AT Failed P=PC input aborted AT 3rd - AT ended in phase L=Listen R=Relay E=Excited adaptive control V=Verification	–
19	ATCCC	system	0	–	**C**ompletion **C**ode of latest **A**uto**T**uning. **C**ounter value when AT ended.	–
20	ATCCD	system	–	–	**C**ompletion **C**ode of latest **A**uto**T**uning. The **D**ate when the autotuning was started.	–
21	ATCCT	system	–	–	**C**ompletion **C**ode of latest **A**uto**T**uning. The **T**ime when the autotuning was started.	–
22	ATCCL	system	0	–	**C**ompletion **C**ode of latest **A**uto**T**uning. **L**ength in minutes.	–

(1) Zx_GAIN=0, indicates that no save has been done for zone x.

Aux control param



Terminal Description, Aux control param

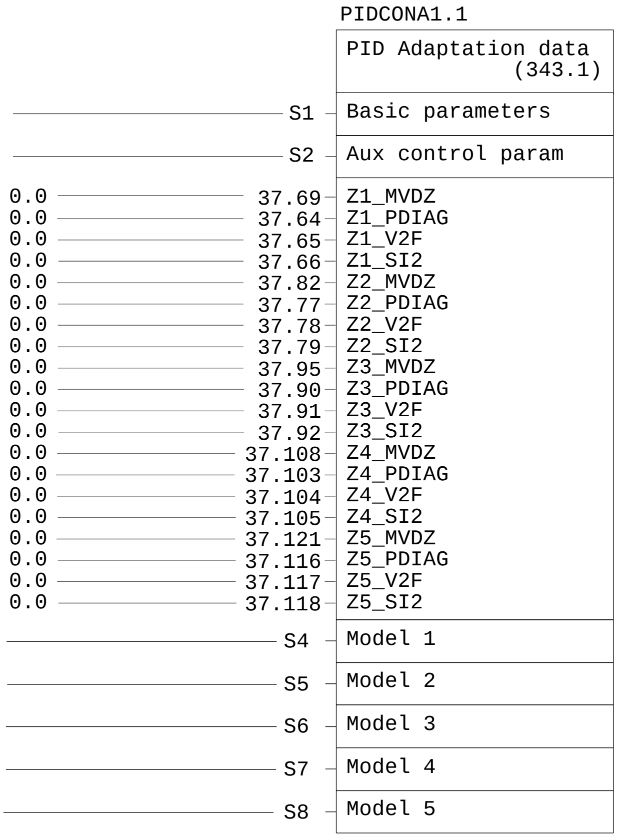
Termi-nal No	Terminal Name	Value entered by	Default value	PC con-nection data type	Description	Remarks
37.61	Z1_BETA	system	1.0	–	**BETA** saved in **Z**one **1**.	≥0
37.62	Z1_TS	system	0.0	–	**TS** saved in **Z**one **1**.	≥0
37.63	Z1_FREQ	system	0.0	–	**FREQ**uency saved in **Z**one **1**.	≥0
37.67	Z1_RESP	system	NORMAL	–	**RESP**onse type saved in **Z**one **1**.	–
37.74	Z2_BETA	system	1.0	–	**BETA** saved in **Z**one **2**.	≥0
37.75	Z2_TS	system	0.0	–	**TS** saved in **Z**one **2**.	≥0
37.76	Z2_FREQ	system	0.0	–	**FREQ**uency saved in **Z**one **2**.	≥0
37.80	Z2_RESP	system	NORMAL	–	**RESP**onse type saved in **Z**one **2**.	–

Terminal Description, Aux control param (Continued)

Terminal No	**Terminal Name**	**Value entered by**	**Default value**	**PC connection data type**	**Description**	**Remarks**
37.87	Z3_BETA	system	1.0	–	**BETA** saved in **Z**one **3**.	≥0
37.88	Z3_TS	system	0.0	–	**TS** saved in **Z**one **3**.	≥0
37.89	Z3_FREQ	system	0.0	–	**FREQ**uency saved in **Z**one **3**.	≥0
37.93	Z3_RESP	system	NORMAL	–	**RESP**onse type saved in **Z**one **3**.	–
37.100	Z4_BETA	system	1.0	–	**BETA** saved in **Z**one **4**.	≥0
37.101	Z4_TS	system	0.0	–	**TS** saved in **Z**one **4**.	≥0
37.102	Z4_FREQ	system	0.0	–	**FREQ**uency saved in **Z**one **4**.	≥0
37.106	Z4_RESP	system	NORMAL	–	**RESP**onse type saved in **Z**one **4**.	–
37.113	Z5_BETA	system	1.0	–	**BETA** saved in **Z**one **5**.	≥0
37.114	Z5_TS	system	0.0	–	**TS** saved in **Z**one **5**.	≥0
37.115	Z5_FREQ	system	0.0	–	**FREQ**uency saved in **Z**one **5**.	≥0
37.119	Z5_RESP	system	NORMAL	–	**RESP**onse type saved in **Z**one **5**.	–

The terminals Z1_RESP, Z2_RESP, Z3_RESP, Z4_RESP and Z5_RESP (the response types saved for the zones) are presented in the same manner as the terminal RESTYPE of the data base element Adaptive PID control.

Aux model param



Terminal Description, Aux model param

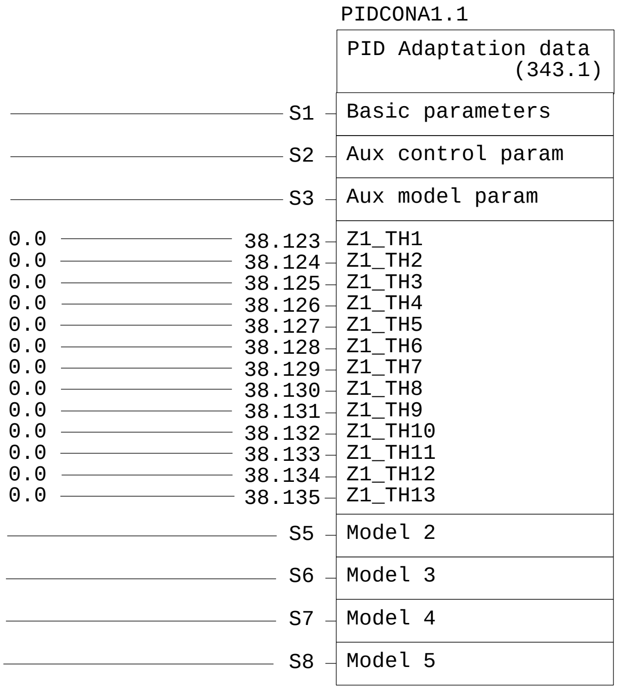
Terminal No	**Terminal Name**	**Value entered by**	**Default value**	**PC connection data type**	**Description**	**Remarks**
37.69	Z1_MVDZ	system	0.0	–	**MV** Adaptation **D**ead**Z**one (%) in **Z**one **1**. No adaptation when the measured value varies less.	≥0
37.64	Z1_PDIAG	system	0.0	–	Adaptation data saved in **Z**one **1**.	≥0 (1)
37.65	Z1_V2F	system	0.0	–	Adaption data saved in **Z**one **1**.	≥0
37.66	Z1_SI2	system	0.0	–	Adaption data saved in **Z**one **1**.	≥0
37.82	Z2_MVDZ	system	0.0	–	**MV** Adaptation **D**ead**Z**one (%) in **Z**one **2**. No adaptation when the measured value varies less.	≥0

Terminal Description, Aux model param (Continued)

Terminal No	**Terminal Name**	**Value entered by**	**Default value**	**PC connection data type**	**Description**	**Remarks**
37.77	Z2_PDIAG	system	0.0	–	Adaptation data saved in **Z**one **2**.	≥0 (1)
37.78	Z2_V2F	system	0.0	–	Adaption data saved in **Z**one **2**.	≥0
37.79	Z2_SI2	system	0.0	–	Adaption data saved in **Z**one **2**.	≥0
37.95	Z3_MVDZ	system	0.0	–	**MV** Adaptation **D**ead**Z**one (%) in **Z**one **3**. No adaptation when the measured value varies less.	≥0
37.90	Z3_PDIAG	system	0.0	–	Adaption data saved in **Z**one **2**.	≥0 (1)
37.91	Z3_V2F	system	0.0	–	Adaption data saved in **Z**one **2**.	≥0
37.92	Z3_SI2	system	0.0	–	Adaption data saved in **Z**one **3**.	≥0
37.108	Z4_MVDZ	system	0.0	–	**MV** Adaptation **D**ead**Z**one (%) in **Z**one **4**. No adaptation when the measured value varies less.	≥0
37.103	Z4_PDIAG	system	0.0	–	Adaption data saved in **Z**one **4**.	≥0 (1)
37.104	Z4_V2F	system	0.0	–	Adaption data saved in **Z**one **4**.	≥0
37.105	Z4_SI2	system	0.0	–	Adaption data saved in **Z**one **4**.	≥0
37.121	Z5_MVDZ	system	0.0	–	**MV** Adaptation **D**ead**Z**one (%) in **Z**one **5**. No adaptation when the measured value varies less.	≥0
37.116	Z5_PDIAG	system	0.0	–	Adaption data saved in **Z**one **5**.	≥0 (1)
37.117	Z5_V2F	system	0.0	–	Adaption data saved in **Z**one **5**.	≥0
37.118	Z5_SI2	system	0.0	–	Adaption data saved in **Z**one **5**.	≥0

(1) Zx_PDIAG=0, indicates that there is no model saved in zone x.

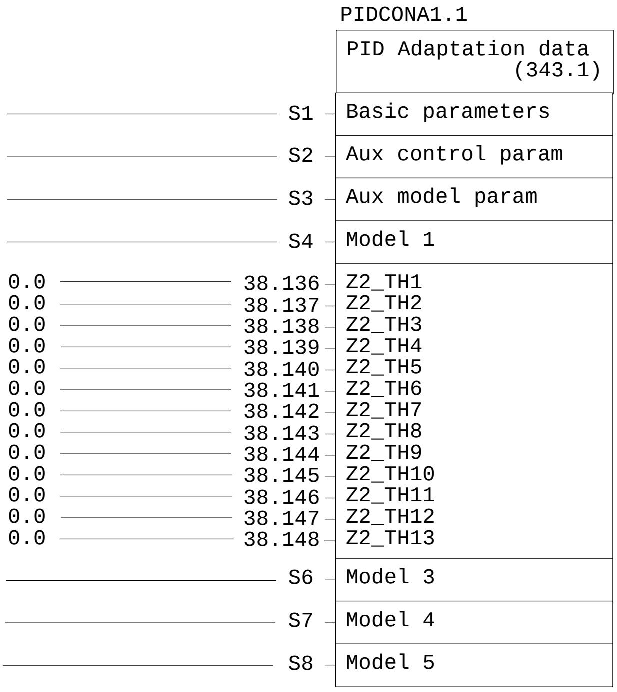
Model 1



Terminal Description, Model 1

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
38.123	Z1_TH1	system	0.0	–	Model value saved in **Z**one **1**.	–
38.124	Z1_TH2	system	0.0	–	Model value saved in **Z**one **1**.	–
:	:	:	:	:	:	:
38.135	Z1_TH13	system	0.0	–	Model value saved in **Z**one **1**.	–

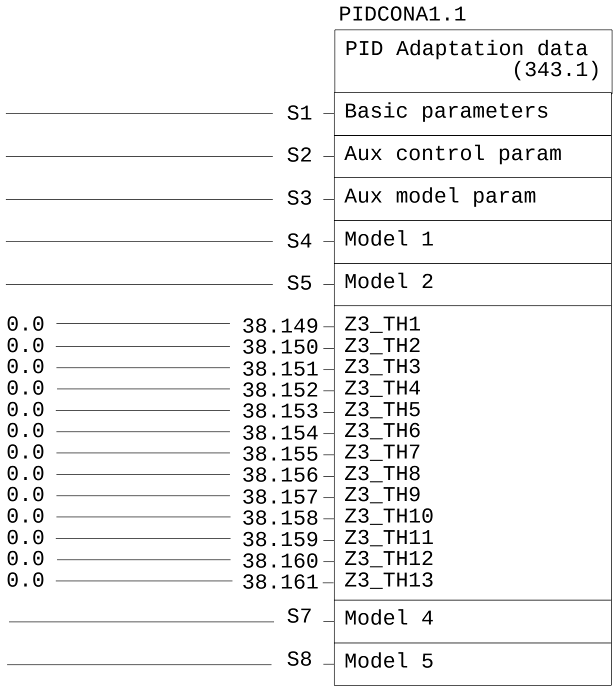
Model 2



Terminal Description, Model 2

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
38.136	Z2_TH1	system	0.0	–	Model value saved in Zone **2**.	–
38.137 : 38.148	Z2_TH2 : Z2_TH13	system : system	0.0 : 0.0	– : –	Model value saved in Zone **2**. : Model value saved in Zone **2**.	– : –

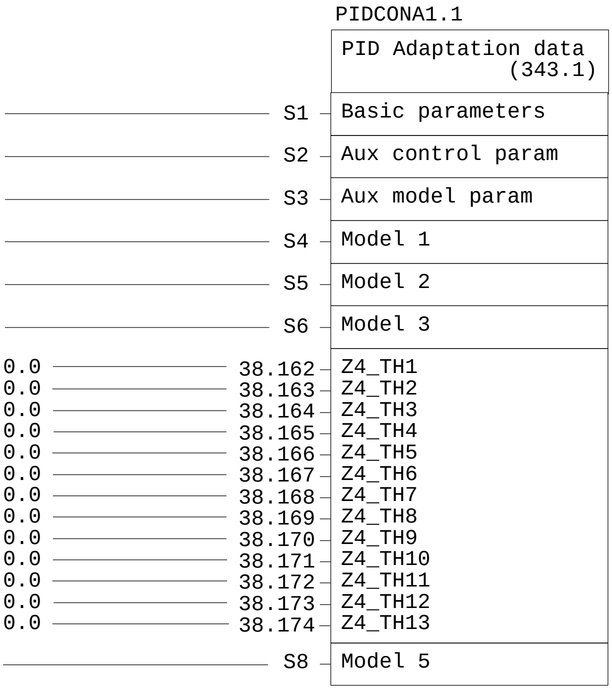
Model 3



Terminal Description, Model 3

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
38.149	Z3_TH1	system	0.0	–	Model value saved in **Z**one **3**.	–
38.150 : 38.161	Z3_TH2 : Z3_TH13	system : system	0.0 : 0.0	– : –	Model value saved in **Z**one **3**. : Model value saved in **Z**one **3**.	– : –

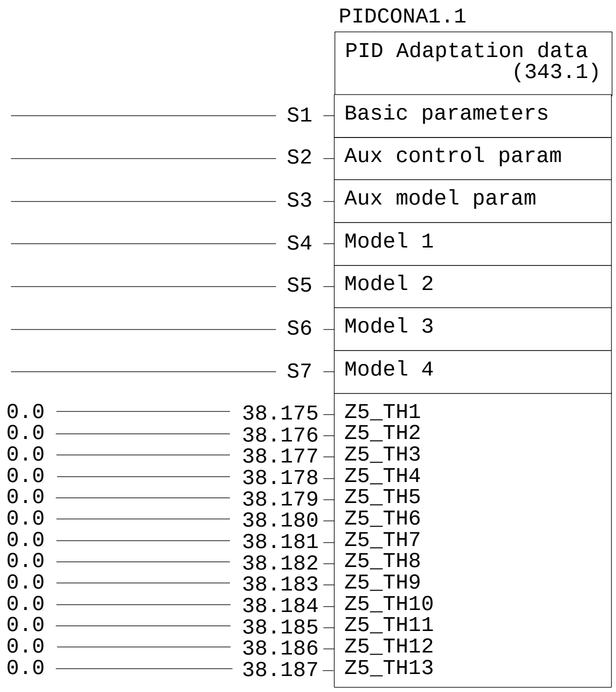
Model 4



Terminal Description, Model 4

Terminal No	Terminal Name	Value entered by	Default value	PC connection data type	Description	Remarks
38.162	Z4_TH1	system	0.0	–	Model value saved in **Z**one **4**.	–
38.163 : 38.174	Z4_TH2 : Z4_TH13	system : system	0.0 : 0.0	– : –	Model value saved in **Z**one **4**. : Model value saved in **Z**one **4**.	– : –

Model 5



Terminal Description, Model 5

Termi-nal No	Terminal Name	Value entered by	Default value	PC con-nection data type	Description	Remarks
38.175	Z5_TH1	system	0.0	–	Model value saved in **Z**one **5**.	–
38.176 : 38.187	Z5_TH2 : Z5_TH13	system : system	0.0 : 0.0	– : –	Model value saved in **Z**one **5**. : Model value saved in **Z**one **5**.	– : –

Appendix E Standard Event Texts

E.1 General

This description of event texts for PIDCONA is reprinted from *AdvaCommand Basic Functions User's Guide*.

E.2 Standard Event Text for Process Controller PIDCONA

Table E–1. Event Texts for PIDCONA

Events	**Event Treat pointer**	**Property text**		**Event text**	
		Standard text	**Line No in Event Treat**	**Standard text**	**Line No in Event Treat**
Lower limit L1 for measured value exceeded	H1L1_TR	MV<L1	1	<Value> <Unit>	1
Lower limit L1 for measured value re-entered	H1L1_TR	MV<L1	1	<Value> <Unit>	2
Upper limit H1 for measured value exceeded	H1L1_TR	MV>H1	2	<Value> <Unit>	1
Upper limit H1 for measured value re-entered	H1L1_TR	MV>H1	2	<Value> <Unit>	2
Lower limit (1) for deviation exceeded	H1L1_TR	DEV<L	3	<Value> <Unit>	1
Lower limit (1) for deviation re-entered	H1L1_TR	DEV<L	3	<Value> <Unit>	2
Upper limit (1) for deviation exceeded	H1L1_TR	DEV>H	4	<Value> <Unit>	1
Upper limit (1) for deviation re-entered	H1L1_TR	DEV>H	4	<Value> <Unit>	2
Lower limit L2 for measured value exceeded	H2L2_TR	MV<L2	1	<Value> <Unit>	1
Lower limit L2 for measured value re-entered	H2L2_TR	MV<L2	1	<Value> <Unit>	2
Upper limit H2 for measured value exceeded	H2L2_TR	MV>H2	2	<Value> <Unit>	1

Table E–1. Event Texts for PIDCONA (Continued)

Events	**Event Treat pointer**	**Property text**		**Event text**	
		Standard text	**Line No in Event Treat**	**Standard text**	**Line No in Event Treat**
Upper limit H2 for measured value re-entered	H2L2_TR	MV>H2	2	<Value> <Unit>	2
PC-gate LOCALFL decides control mode	MODE_TR	LOCAL !!	1	On	1
PC-gate MANFL decides control mode	MODE_TR	MAN !!	2	On	1
PC-gate AUTOFL decides control mode	MODE_TR	AUTO !!	3	On	1
PC-gate E1 decides control mode	MODE_TR	E1 !!	4	On	1
PC-gate E2 decides control mode	MODE_TR	E2 !!	5	On	1
PC-gate E3 decides control mode	MODE_TR	E3 !!	6	On	1
AI-error On	ERR_TR	AISign Err	1	Alarm	1
AI-error Off	ERR_TR	AISign Err	1	Normal	2
Autotuning aborted	ADAP_TR	AT Abort	1	Alarm	1
Autotuning aborted	ADAP_TR	AT Abort	1	Normal	2
Autotuning failed	ADAP_TR	AT Fail	2	Alarm	1
Autotuning failed	ADAP_TR	AT Fail	2	Normal	2
Autotuning. Question to operator	ADAP_TR	AT Quest	3	Decision	3
Autotuning. Question to operator answered	ADAP_TR	AT Quest	3	Answered	4
There is an invalid sample rate	ADAP_TR	TS Fault	4	Alarm	1
There is an invalid sample rate	ADAP_TR	TS Fault	4	Normal	2
Adaptation failed	ADAP_TR	Ad Fail	5	Alarm	1
Adaptation failed	ADAP_TR	Ad Fail	5	Normal	2

Appendix F Dominant Pole Placement, the PID Tuning Method

F.1 Introduction

The purpose of this text is to give some theoretical background for the tuning done in PIDCONA. Thereby we hope to shed some light over the mystery of how it uses a discrete time model of the process to get PID controller parameters that give the desired behavior of the closed loop. The text is mostly directed towards those of you who have education in automatic control, enough to be familiar with such notions as poles and zeros of a transfer function.

Some examples, with different types of processes, are included to illustrate how the tuned closed loop responds to disturbances and setpoint steps, in comparison to the uncontrolled process.

F.2 The Basic Idea of Dominant Pole Placement

The poles of the transfer function of the closed loop carry a lot of information about the behavior of the loop. There is often a dominant pole pair. For a continuous time transfer function it is likely to be placed as in Figure F-1. Assuming that the other poles are distinctly faster (or are placed very close to zeros of the transfer function), you can deduce the shape and speed of the settling transients if you know these dominant poles.

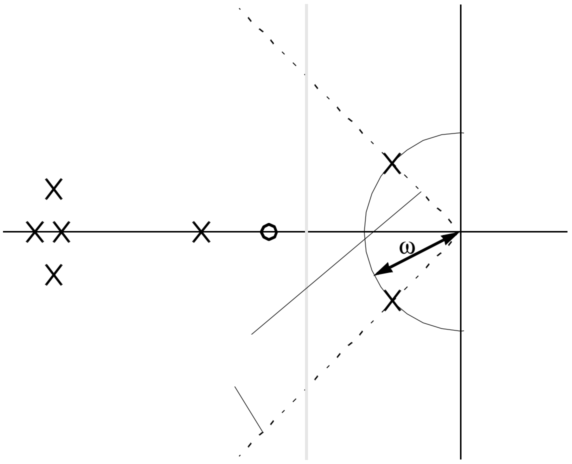


Figure F-1. A typical closed loop configuration of poles and zeros of a continuous time transfer function (X = pole, O = zero)

The dominant part of the transfer function illustrated in Figure F-1.

$$\frac{K\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

ζ is the relative damping and ω is the natural frequency associated with the dominant poles. It is a second order transfer function. Its two poles are complex if $0<\zeta<1$. For a given value of the relative damping ζ, the poles lie on the dashed lines of Figure F-1, and their distance to the origin equals the frequency ω. If these poles are moved along the dashed lines (ζ is constant), the transfer function gets quicker the further away from the origin the poles are placed (higher ω value). This does not change the shape of the step response for this dominant part, only its time scale, as shown in Figure F-2. Thus the speed is determined by the frequency ω.

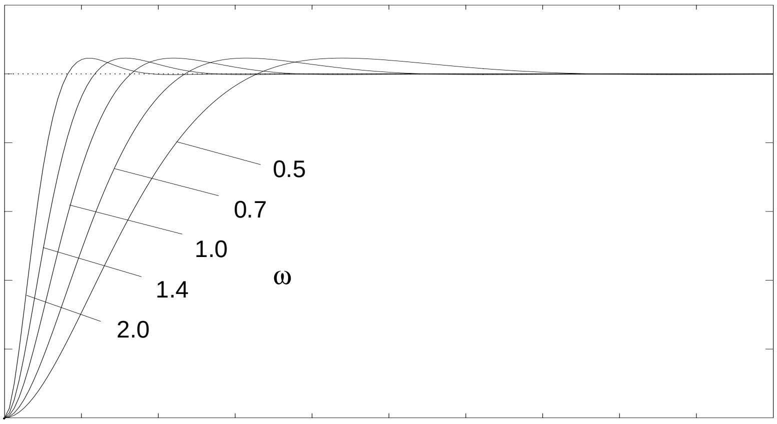


Figure F-2. Step responses for a second order transfer function with relative damping $\zeta = 0.7$ and different natural frequency ω

If, instead, the relative damping ζ is varied between 0 and 1, keeping the frequency ω constant, the poles move on the half circle shown in Figure F-1. For damping values above 1, the poles are on the negative real axis. The effect of different damping values on the step response is shown in Figure F-3. The step response is more damped the higher the relative damping ζ is.

The strong coupling between the location of the poles and the behaviour has motivated the development of controller design methods where the closed loop poles are specified.

With a PI or PID controller it is possible to place two or three poles of the closed loop at desired locations. In total, the closed loop often has far more poles than that. Therefore, standard pole placement (in which all closed loop poles are specified) is in general not feasible for tuning of PID controllers.

The idea of dominant pole placement is to place dominant poles where they give the desired behaviour. Note that this statement says more than just placing two or three of the poles at suitable locations. It also requires that these placed poles do in fact dominate over the other poles.

With a process model available and the controller structure defined (PI or PID in our case), it is a straight-forward task to calculate the controller setting (gain, integral time, and possibly derivative time), once the dominant poles have been decided.

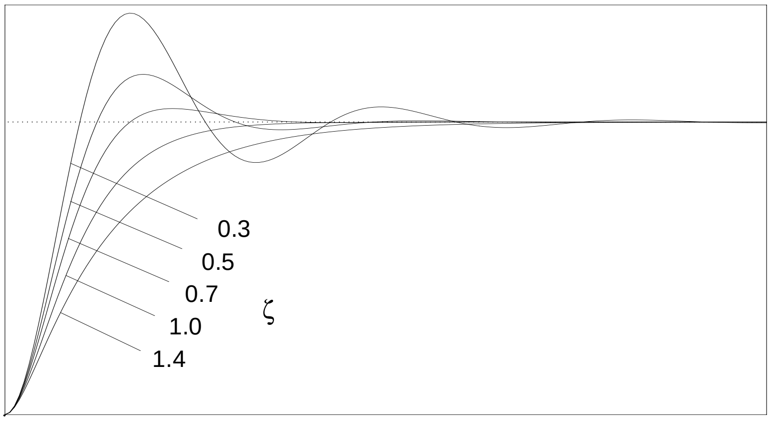


Figure F-3. Step responses for a second order transfer function with different relative damping ζ but the same natural frequency ω

F.3 Dominant Pole Placement in PIDCONA

The desired behavior can be specified by giving the relative damping for the dominant pole pair. For PIDCONA this is done by selecting one out of five response types, based on type curves for step responses.

Different applications put different demands on how the process should settle after a disturbance. It is mostly better the quicker it settles, but in some cases it is important to avoid overshoots, and in other cases overshoots make no harm as long as a smoothed-out curve returns to the setpoint as quickly as possible. This is why it is left to the user to decide which type of response to aim at, classified by its relative damping. The character of the transient caused by a disturbance resembles the step response very much.

Consider Figure F-1 again. As long as only the relative damping is specified but not the natural frequency, the specified pole pair is free to move along the dashed lines. When making it faster, the remaining poles usually get slower, i. e. they come closer to the imaginary axis. If made too fast, the pair no longer dominates over the other poles. Then the specified poles are no longer dominant. The visualization in Figure F-1 assumes that the specified pole pair is complex, $\zeta<1$, but the reasoning holds just as well in cases with higher relative damping values and thus real specified poles.

Figure F-1 could make you believe that the dominant poles are always distinctly slower than the others. This is, of course, the normal criterion for dominance. But if there are slow process zeros, they will attract closed loop poles when the controller gain is increased, and when really close these poles will have little effect on the closed loop behavior, compared to the ones that are far from such zeros. For a group of poles to be dominant, they shall with some margin be the slowest poles that are not very close to process zeros.

PIDCONA makes a pair of poles, with the specified relative damping, as fast as possible with retained dominance. In the PID case (as opposed to PI), a third pole may be used as well, placed in between the other two and neutralized in the step response by a zero at the same place. Figure F-4 shows typical results of the dominant pole placement used in PIDCONA. The example used is a second order process with deadtime.

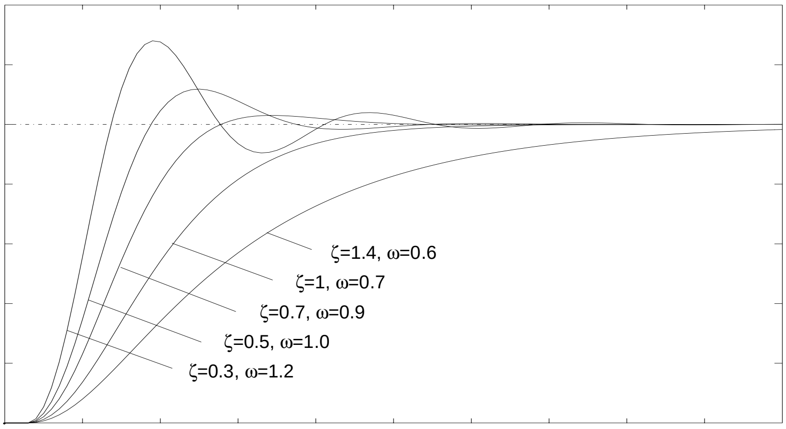


Figure F-4. Typical closed loop step responses with PIDCONA’s dominant pole placement. The highest natural frequency ω giving dominance is used for each specified relative damping ζ.

Since sampled control is used, there is a sampling rate involved. The calculations would not be accurate enough using continuous time models and pole locations, unless the sampling rate was unnecessarily high. PIDCONA therefore uses discrete time models and the discrete time counterparts of the continuous time pole locations discussed above.

The models used always have some uncertainty. This is taken into account in a robustness test, making sure that the closed loop is not made faster than permissible with respect to the model uncertainty.

There is also always the possibility for the user to limit the natural frequency, ω, of the dominant poles, so that the design is not made too fast with respect to actuator ware, high frequency noise, or whatever reason there could be for restricting the speed.

You could see, in Figure F-3, that there is a difference in speed for different values of the relative damping, ζ, even if the frequency, ω, is the same. If you compare this with the step responses in Figure F-4, you notice that the difference in speed gets greater when the frequency, ω, is maximized as done in PIDCONA.

F.4 Examples

In order to illustrate the behavior obtained, this section shows examples where the dominant pole placement design method of PIDCONA has been applied to processes of some different types. Trend curves have been generated by simulations, where the process was at rest initially. After a while there is a load disturbance in the form of a step, parallel to the manipulated variable, and some time later a setpoint step is made. In Figures F-5 through F-8 the solid lines show the controller output (the manipulated variable) and the measured value (the controlled variable), for the closed loop. Dashed lines show what the response to the step disturbance would have been without the control.

The first example is a second order process with a short deadtime. The deadtime is 0.6 and the time constants are 2 and 4. (The same process was in fact used for the generation of Figure F-4.) The response type is specified as ‘Normal’, which corresponds to the relative damping ζ=0.7. The control obtained is shown in Figure F-5.

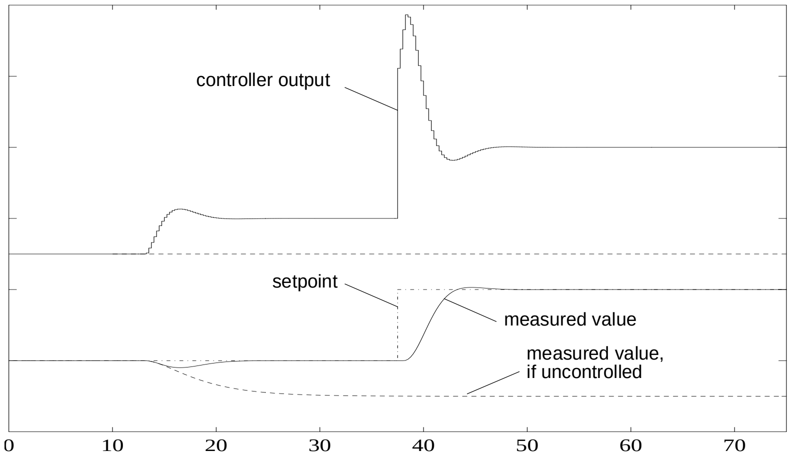


Figure F-5. A second order process with a short deadtime. Load disturbance step at time 12.5. Setpoint step at time 37.5.

The second example has significantly longer deadtime. It is also a second order process with the same time constants as the previous example, but the deadtime is 8. Response type 'Normal' is used, and the resulting behavior is seen in Figure F-6.

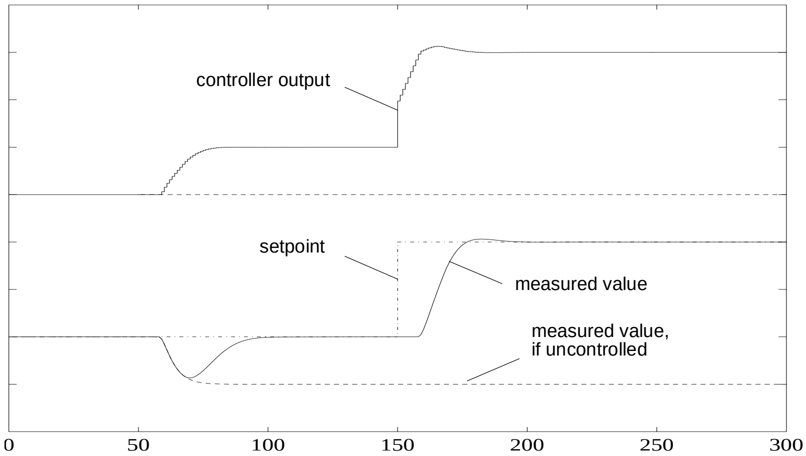


Figure F-6. A second order process with significant deadtime. Load disturbance step at time 50. Setpoint step at time 150.

In the next example the process has non-minimum phase behavior. Its response to a change of the manipulated variable has an initial direction opposite to the final response. The response type 'Fast' is specified. Figure F-7 shows the result.

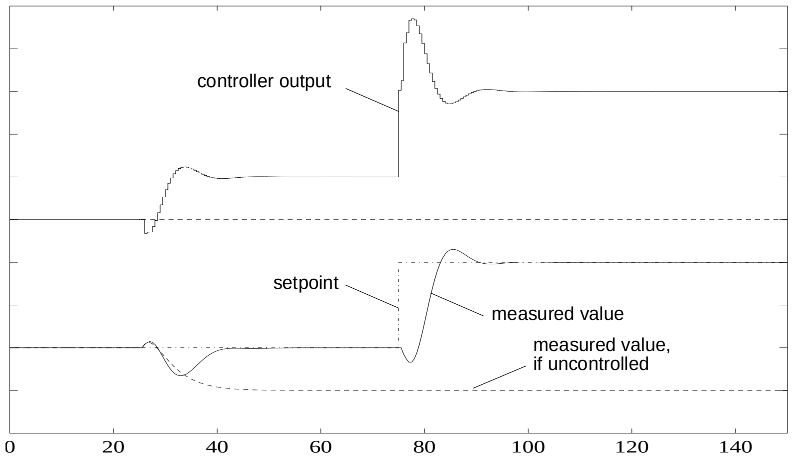


Figure F-7. A non-minimum phase process. Load disturbance step at time 25. Setpoint step at time 75.

Finally an example is presented where the open loop process is unstable. It emanates from a chemical reactor with an exothermal reaction, where the temperature is controlled by varying the cooling. Beside the unstable mode, there is also a deadtime in the process. The tuning is made with response type 'Normal', giving the results shown in Figure F-8. Since increased cooling has a decreasing effect on the temperature, direct action is used. (In the previous examples, we had reverse action.)

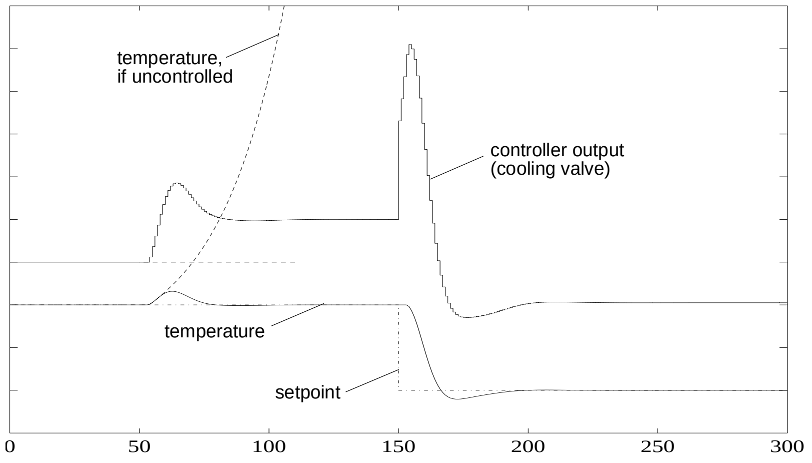


Figure F-8. Temperature control of an exothermal chemical reaction. Load disturbance step at time 50. Setpoint step at time 150.

F.5 Model of the Closed Loop

When dominant pole placement is used, it is simple to find a model for the closed loop. The dominant part of the transfer function is defined by the dominant poles. There may be process zeros and a deadtime, that also have significant influence over the response. But essentially the dominant poles hold the information needed to get a reasonable model of how the process is expected to react to changes of the setpoint.

The model, fed with the setpoint as input, produces an expected response, which can be used in the supervision of the loop. In PIDCONA (if autotuned) the deviation supervised is the difference between the measured value and this expected response.

The benefits are illustrated by Figure F-8, which shows an actual recording from a reactor of the type used in Figure F-8. The PID controller was tuned by dominant pole placement, and model based supervision was used. The setpoint is ramped from one temperature to another with high restrictions on the deviation. It is obvious that the model based supervision can use tighter limits, without causing excessive alarms, than if the direct difference between the measured value and the setpoint is used.

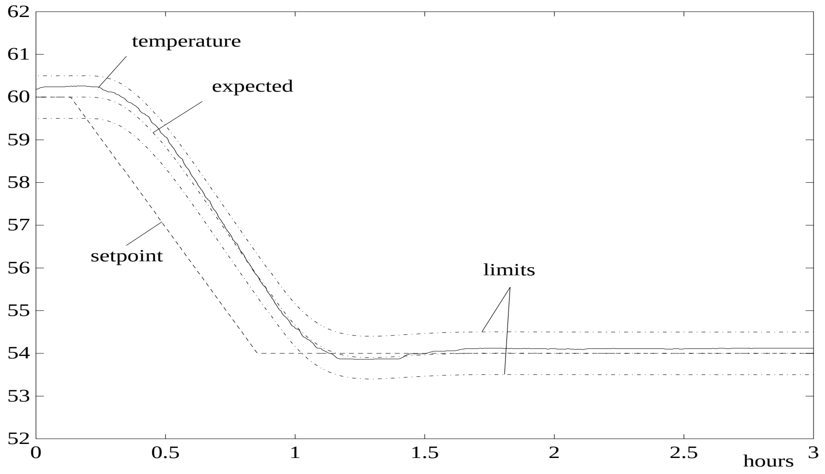


Figure F-9. Model based supervision

INDEX

A
ABL_H1L1 3-8, 4-20
ABL_H2L2 3-8, 4-20
Accept & Save 4-34
ACT_DIR 3-7
ACTPOS 3-7
actuator
    deadzone 4-37
    fault 5-4
    limit 4-33
    position 3-7, 3-17
    rate limit 3-6, 3-12, 4-33, 5-3
    wear 4-33
Ad Fail 4-29, 5-4
ADAPON 3-7, 3-13, 4-18
adaptation
    deadzone 4-34, 4-37, 5-5
    Enable/Off 4-40
adaptive control
    description 1-3, 1-6, 3-7, 4-5, 4-18
    precautions 3-5 to 3-7
    problems 4-29, 5-4
    selecting 4-18, 4-40
ADDOUT 3-7 to 3-8, 4-1, 4-5, 4-10
AI error alarm 4-2 to 4-3, 4-20, 4-29
AISign Err 4-29
alarm
    blocking 4-20, 4-30
        programmed 4-20
    list 4-20
    text 4-29
    types 4-20, 4-30
alarm blocking
    programmed 3-8, 3-13
ALCBLK 3-8, 4-20
aliasing 3-2
analog input error
    alarm 4-2 to 4-3, 4-20, 4-29
    channel
        AI_ERR 3-17
anti reset wind-up 3-6, 4-5
anti-alias filter 3-2
AT
    Abort 4-29
    Fail 4-29
    Quest 4-29, 4-34
ATENBL 3-3, 3-13
AUTOENBL 3-12, 4-3
AUTOFL 3-5, 4-2, 4-20
AUTOSP 4-1, 4-7, 4-27
    tracking 4-7, 4-9
autotuning 1-3, 1-6, 4-13, 4-31
    alarms 4-29
    control modes 4-3
    fault finding 5-2
    for gain scheduling 4-17, 4-39
    tutorial 4-22
    with feedforward 3-7

B
balance mode
    BAL 3-2
BALIN 3-4, 4-2 to 4-3
BALREF 4-1, 4-10
BETA
    setpoint factor 4-4, 4-37 to 4-38
blocking
    alarm 3-8, 4-20, 4-30
BUMPLESS 3-4, 3-9, 4-5, 4-8, 4-11
bumpless transfer 1-7, 3-3, 3-9, 4-5, 4-10, 4-12

C
cascade control 3-3 to 3-4
CLAMP 4-2 to 4-3
clamped control 3-2
CLAMPREF 4-1, 4-10
class 3-17
Conditioned transfer 4-11
CON-PU1 4-37
Continue Tune 4-17, 4-31, 4-34, 4-39
control algorithm 4-4
control modes
    configuration 3-1 to 3-2, 3-4, 3-9
    description 4-1, 4-8, 4-10
    events 4-20
    optional names 3-17
    selecting 4-2 to 4-3, 4-26
    when autotuning 4-13 to 4-14, 4-23, 4-31, 5-2
controlled variable
    see measured value 1-4
Controller 1-1 to 1-2
controller 1-7, 4-1, 4-4
controller output
    see output 1-4

critically damped 4-33
cycle time 3-1, 5-3
  change 4-30, 5-2
  example 3-12
  for output ramping 4-10
  in cascade control 3-5
  versus controller sampling period 4-5

D
data base element 3-11, 3-15, 3-17
  subordinate 3-17
data base source code 3-17
DBINST 1-9
deadzone
  DZ 4-37
decimals 3-17
Decr 4-22
decrease key 4-27
derivative time
  TD 4-4, 4-37
DERIVDEV 3-8, 4-5
DEV 4-4
DEVHYST 4-20
deviation 3-9, 4-4
  alarm 4-20
  limits 4-28
direct action 3-6, 4-4, 4-22
disturbance 3-1, 3-7, 4-25, 5-2, 5-4
  rejection 4-36
DZ
  deadzone 4-37

E
E1 mode 3-4, 4-1, 4-8
E1ENBL 4-3
E1FL 4-2, 4-20
E2 mode 3-4, 4-1, 4-8
E2ENBL 4-3
E2FL 4-2, 4-20
E3 mode 3-2, 4-10
E3ENBL 4-3
E3FL 4-2, 4-20
EOLIM 3-5, 4-10
EOLL, EOHL 3-5
error codes 5-1
event
  list 4-20
  texts 4-29
  treat pointers 4-20, 4-29
  types 4-20, 4-30
Excitation level 4-25, 4-33
Excitation phase 4-14, 4-23
execution time 3-9
Extended Setu 4-35
Extended Setup 4-22, 4-33
extension 3-11, 3-18
external output
  limits 3-5, 3-18
external output limits 4-10
EXTREF1 4-1, 4-8
EXTREF1, EXTREF2 3-4
EXTREF2 4-1, 4-8
EXTREF3 3-3, 4-1, 4-10

F
fast process 3-1
feedforward 3-7
filter
  anti-alias 3-2
Frequency
  Limit 4-19, 4-33, 4-36, 5-3 to 5-4
  Unlimited 4-33

G
Gain 4-4, 4-37
gain scheduling 1-4, 1-6, 3-7, 4-16, 4-38, 5-4 to 5-5
  adaptive control with 4-18
  autotuning with 4-39
  example 3-14
  parameter sets 4-18
  range 3-17
group alarm 3-8, 4-29
GSENBL 3-16, 4-16 to 4-17, 4-40

H
hysteresis
  alarm limit 3-17, 4-20
  relay 4-13, 5-3
  zone limit 3-7, 4-6, 4-16

I
Incr 4-22
increase key 4-27
integration time
  TI 4-4, 4-37
ISE 4-33
ITAE 4-33

L

- limit
  - alarms 4-20
- limit alarms
  - blocking 4-20, 4-30
  - hysteresis 3-17
  - levels 4-28
  - programmed blocking 3-8
- limits 4-6, 4-21
  - output 3-6 to 3-7, 4-5, 4-10
    - external 3-5, 3-18, 4-10
    - in cascade control 3-5
    - when autotuning 4-14, 4-33
  - setpoint 4-7
- Listen phase 4-13, 4-23
- load 3-1, 3-9
- LOCAL 4-2 to 4-3
- LOCALFL 4-2 to 4-3, 4-20
- Low Excitation 4-25

M

- MAN Clamped 3-2, 4-1, 4-10
- Man Forced 4-2 to 4-3
- MANENBL 3-5, 3-12, 4-3, 4-27
- MANFL 4-2, 4-20
- manipulated variable
  - see output 1-4
- MAX 4-6
- measured value
  - MV 1-4, 4-4
- memory requirement 3-9
- MIN 4-6
- model 4-13, 4-18, 4-37
- model based supervision 4-20
- modes
  - see control modes 4-1
- MVAAF 4-6, 4-20
- MVHYST 4-20

N

- number of decimals 3-17
- Nyquist frequency 3-2

O

- object display 1-6, 4-21
  - present and saved parameters 4-19
  - when autotuning 4-23 to 4-25
  - with gain scheduling 4-39
- object trend 3-18
- OUTMAX 4-6
- OUTMIN 4-6
- output
  - generation 4-4, 4-10
  - limits 3-6 to 3-7, 4-10
    - external 3-5, 3-18
  - OUT 1-4
  - ramping 4-10
  - rate limit 3-6, 4-10, 5-3
  - source of 4-1
- output limits 4-5
  - external 4-10
- output rate
  - limit 3-12
- overdamped 4-33
- overshoot 4-32

P

- parameter scheduling
  - see gain scheduling 1-4
- physical units 4-6
- PI control 4-37
- Present parameters 4-4, 4-18
- process model 4-13, 4-18, 4-37
- process value
  - see measured value 1-4
- processing load 3-1, 3-9
- proportional control 3-8, 4-5, 4-8, 4-37

Q

- quarter amplitude damping 4-33

R

ramping
    output 4-10
    setpoint 4-8
ranges 3-17, 4-6
rate limit
    actuator 3-12, 4-10, 4-33
    output 3-12, 4-10, 4-33
        when autotuning 5-3
    output and actuator 3-2, 3-6
        with feedforward 3-7
    setpoint 3-16, 4-7
    setpoint, when autotuning 4-14
Relay phase 4-13, 4-23
reset wind-up 3-6, 4-5
Response Setup 4-22, 4-32, 4-35
RESTORE 3-7, 4-18
Restore 4-19, 4-41
restore 4-19
Retrieve aborted setting 4-34 to 4-35
retuning 4-31, 5-1
REVACT 3-6, 3-12, 4-4, 4-22
reverse action 3-6, 4-4, 4-22
robustness 4-14, 4-35, 5-4

S

sampling period
    sampling rate 3-1 to 3-2, 3-9
    TS 4-5, 4-37
Save 4-19, 4-37, 4-41
saved models 3-17
Saved parameters 3-17, 4-18, 4-38
scaling 3-7, 4-6
SCHEDIN 3-16, 4-6, 4-16, 4-40
SCHMAX 4-6
SCHMIN 4-6
section 3-17
sensitivity function 4-35
SEQAUTO 3-5, 4-2
SEQE1 4-2
SEQE2 4-2
SEQE3 4-2
SEQMAN 4-2
setpoint 1-4
    generation 4-7
    indication 4-27 to 4-28
    optimization 3-5
    ramping 4-8
    source of 4-1
    tracking 4-7, 4-9
        example 3-4
setpoint factor
    BETA 4-4, 4-37 to 4-38
setpoint tracking 3-9
    example 3-3, 3-12
settling time 3-1, 4-32
SHOW_ACT 3-7
slow process 3-1
source code
    data base 3-17
SPEEDA 3-12, 3-16, 4-8, 4-11
SPEEDE1 4-8
SPEEDE2 4-8
SPEEDO 3-2, 3-6, 3-12, 3-16, 4-10, 4-28
Start Tune 4-23, 4-31, 4-39
status list 3-17
step response 4-32
Stop & Restore 4-34
subordinate data base element 3-17
system messages 5-1

T

TD
    derivative time 4-4, 4-37
three-point controller 4-37
TI
    integration time 4-4, 4-37
TRACKEXT 3-4, 4-9, 4-27
tracking
    output 4-10
    setpoint 3-9, 4-7, 4-9
        example 3-4, 3-12
    setpoint, example 3-3
TRACKMV 3-3 to 3-4, 3-9, 3-12, 4-9, 4-27
TS
    Fault 4-29, 5-2
    sampling period 4-5, 4-37
TTDLOG 3-18
tuning setup 4-35
two degrees of freedom 4-38

U

units 3-7, 3-17, 4-5 to 4-6

V

Verification phase 4-15, 4-24, 4-34
Verify 4-34

W

wind-up 3-6, 4-5
working setpoint
    WSP 1-4, 3-9, 4-7
WSP
    tracking 3-4

Z

ZLIM12 3-16, 4-16
ZLIM23 3-16, 4-16
ZLIM34 3-16, 4-16
ZLIM45 3-16, 4-16
zone
    gain scheduling 1-6, 4-16, 4-38
Zone Info 4-19, 4-38, 4-40
zone limit
    crossing 4-16, 4-18, 4-40
    hysteresis 3-7, 4-6, 4-16
    indication 4-38
zone limits
    example 3-16

3BSE 002 433R0101

February 1998